

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

REGULÁTOR TEPLoty S ESP8266

TEMPERATURE REGULATOR WITH ESP8266

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Vymazal

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Ivo Lattenberg, Ph.D.

BRNO 2021

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Tomáš Vymazal

ID: 214411

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Regulátor teploty s ESP8266

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku elektronického měření teploty. Navrhněte a vyrobte elektronický regulátor teploty s ESP8266. Ten bude spínáním externího topení (230V, max 16A) udržovat teplotu ve stanoveném intervalu. Uvažujte rozsah teplot alespoň -20 až 50 stupňů Celsia. Zařízení bude napájené z externího zdroje. Regulátor bude pracovat ve dvou WiFi režimech. Prvním bude AP mód, kde prostřednictvím webového rozhraní bude možno regulátor konfigurovat, bude zde také vidět průběh teplot a stav sepnutí externího topení. V druhém režimu (Client mód) bude regulátor odesílat naměřená data na cloudovou službu pro IoT analýzu dat typu ThingSpeak apod.

DOPORUČENÁ LITERATURA:

[1] BRTNÍK, Bohumil a David MATOUŠEK. Mikroprocesorová technika: [práce s mikrokontroléry řady ATMEL AVR ATXmega A4]. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-406-4.

[2] MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. μ C & praxe. ISBN 80-7300-209-4.

Termín zadání: 1.2.2021

Termín odevzdání: 31.5.2021

Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Bakalářská práce se zabývá vývojem chytrého zásuvkového termostatu, který je schopen komunikovat s uživatelem a dalšími aplikacemi pomocí wifi rozhraní v pásmu 2,4 GHz. Související problematikou je elektrické měření teploty. Další oblastí, na kterou se práce zaměřuje, jsou IoT online cloudy, které se v dnešní době používají pro logování a vizualizaci naměřených dat. V praktické části je podrobně rozebrán vývoj DPS od návrhu až po jednotlivé vývojové verze. Neodmyslitelnou součástí je také firmware a ověření funkčnosti regulace.

Klíčová slova

ESP8266, DPS, teplotní čidla, termostat, regulace, IoT, ThingSpeak, Arduino

Abstract

The bachelor thesis mainly focuses on a development of a smart socket thermostat able to communicate with a user and other applications via Wi-Fi interface in the 2.4 GHz band. Subsequent centers of attention are electrical temperature measurement and analysis of IoT clouds, currently used for logging and visualization of gauged data. Essential part of the work is the detailed depiction of PCBs development from their design to the particular stages of progress. The last but not least are the firmware measures and the regulation functionality verification.

Keywords

ESP8266, PCB, temperature sensors, thermostat, regulation, IoT, ThingSpeak, Arduino

Bibliografická citace

VYMAZAL, Tomáš. *Regulátor teploty s ESP8266*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/133402>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

Prohlášení autora o původnosti díla

Prohlašuji, že svou bakalářskou práci na téma „Regulátor teploty s ESP8266“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 30. května 2021

podpis autora

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Ivo Lattenbergovi, Ph.D., za odborné vedení, trpělivost, užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne: 30. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. PROBLEMATIKA MĚŘENÍ TEPLOTNÍMI SENZORY	12
1.1 ANALOGOVÉ TEPLOTNÍ SENZORY [1].....	12
1.1.1 Kovová čidla	12
1.1.2 Polovodičová teplotní čidla	13
1.2 DIGITÁLNÍ TEPLOTNÍ SENZORY	16
1.2.1 DS18B20 [6].....	16
1.2.2 AM2302 [7].....	17
1.3 TERMOČLÁNKY [1].....	18
2. ROZBOR IOT CLOUDŮ	19
2.1 THINGSPEAK.....	20
2.2 TMEP [10].....	21
2.3 OPENHAB	22
2.4 MQTT.....	22
3. HARDWARE.....	23
3.1 MIKROKONTROLÉR A JEHO PERIFERIE	24
3.1.1 ESP8266.....	25
3.2 OVLÁDÁNÍ SPÍNANÉ ZÁSUVKY	26
3.3 PŘEVODNÍK Z USB NA UART.....	28
3.4 I2C EEPROM PAMĚŤ	29
3.4.1 I2C	29
3.5 VÝROBA A NÁVRH DPS.....	30
3.6 OSAZENÍ A OŽIVENÍ DPS	34
3.6.1 Deska zařízení verze 1.....	34
3.6.2 Oživení zařízení verze 1	35
3.6.3 Deska zařízení verze 2.....	36
3.6.4 Oživení zařízení verze 2	37
3.6.5 Deska zařízení verze 3.....	39
3.6.6 Oživení zařízení verze 3	40
4. FIRMWARE.....	42
4.1 POUŽITÉ KNIHOVNY	44
4.2 VYTVOŘENÉ KNIHOVNY	46
4.3 ZÁVISLOSTI MEZI TŘÍDAMI	47
5. ROZHRANÍ ZAŘÍZENÍ	48
5.1 WEBOVÉ ROZHRANÍ	48
5.2 SÉRIOVÁ LINKA	49
5.3 SIGNALIZAČNÍ DIODY	50

6. OVĚŘENÍ FUNKČNOSTI REGULACE	51
7. ZÁVĚR.....	52
LITERATURA.....	53
SEZNAM SYMBOLŮ A ZKRATEK	55
SEZNAM PŘÍLOH.....	56
OBSAH PŘILOŽENÉHO ARCHÍVU	58

SEZNAM OBRÁZKŮ

1.1	Teplotní senzor PT100	13
1.2	NTC termistor 640-10K	14
1.3	NTC termistor B57164K0472K000	14
1.4	Senzor KTY81 110.....	15
1.5	Jednoduchý způsob odečítání hodnot z analogového teplotního čidla	15
1.6	Teplotní senzor DS18B20 [6].....	16
1.7	Graf odchylky při měření teploty pomocí DS18B20.....	17
1.8	Graf odchylky při měření teploty pomocí AM2302 [7]	17
1.9	Graf odchylky při relativní vlhkosti pomocí AM2302 [7]	18
1.10	Tabulka přehledu termočlánků [8]	18
2.1	Ukázka z uživatelského rozhraní ThingSpeak [13].....	20
2.2	Mapa čidel vyexportovaná 6.12.2020 v 17:15 [10].....	21
3.1	Blokové schéma zapojení MCU a jeho periférií	24
3.2	Schéma desky modulu ESP [18]	25
3.3	Schéma zapojení obvodu pro spínání výstupní zásuvky	26
3.4	Schéma zapojení převodníku z USB na UART.....	28
3.5	Schéma zapojení EEPROM paměti.....	29
3.6	3D model navržené desky verze 1, export KiCad	31
3.7	3D modely DPS verze 2 přední pohled, export KiCad.....	32
3.8	3D modely DPS verze 2 zadní pohled, export KiCad	32
3.9	3D model DPS verze 3 přední pohled, export KiCad.....	33
3.10	3D model DPS verze 3 zadní pohled, export KiCad	33
3.11	Vyrobená deska verze 1 - přední strana	34
3.12	Vyrobená deska verze 1 - zadní strana.....	35
3.13	Vyrobená a osazená deska verze 2 – přední strana	36
3.14	Vyrobená a osazená deska verze 2 – zadní strana.....	37
3.15	Speciální distanční sloupek pro propojení desek.....	38
3.16	Vyrobená a osazená výpočetní deska verze 3 – přední strana.....	39
3.17	Vyrobená a osazená výpočetní deska verze 3 – zadní strana	40
4.1	Diagram popisující základní strukturu firmwaru [21].....	43
4.2	Struktura popisující závislosti mezi použitými třídami [22]	47
5.1	Úvodní stránka webového rozhraní.....	48
5.2	Výpis akcí provedených zařízením vypsaných v terminálu	49
6.1	Graf nespojitě regulace vytvořený v programu Excel	51

SEZNAM TABULEK

2.1	Tabulka nejpoužívanějších IoT cloudů	19
3.1	Tabulka potřebných napěťových úrovní pro ESP8266	24
3.2	Tabulka změřených pracovních bodů LED diod	41
3.3	Tabulka zapojení teplotního čidla k RJ konektoru	41
5.1	Signalizace LED diodami.....	50

ÚVOD

Zadáním bakalářské práce je vývoj a výroba elektrického regulátoru teploty s určitými parametry, jako jsou například možnost připojení zařízení do internetové sítě nebo schopnost spínat síťové napětí 230 V s proudem 16 A. Zařízení vyvíjené v této práci se často označuje jako elektronický termostat. Jedná se o zařízení, které je schopné spínat topení a docílit tak požadované/nastavené teploty na termostatu. Regulátorem teploty se tedy rozumí zařízení, které obsahuje jak část termostatu, tak samotné topné těleso. Tato práce se zabývá pouze návrhem části termostatu, jako topné zařízení se zde uvažuje libovolné elektrické topné zařízení, které lze připojit do zásuvky a je schopno vytápět nějaký prostor v okolí termostatu.

Nedílnou částí této práce je také analýza potřebné teorie ohledně měření teploty, způsobu měření a rozboru typů jednotlivých čidel, které se používají v elektrotechnice pro měření teploty. Problematika klade důraz na využitelnost pro vyvíjené zařízení tak, aby bylo schopno měřit teplotu v intervalu -20 až 50 °C.

Dalším tématem, kterým se práce zabývá je problematika IoT cloudu jakožto možného uložistiště dat získaných z termostatu. Výběr vhodného cloudu se nesmí podcenit, protože v této době existuje mnoho služeb, které by se daly k tomu účelu použít. Každé, pro projekt použitelné uložistiště, má specifické požadavky na komunikaci se zařízeními, tudíž komplikuje možnost migrace do jiné služby. Jediným a častým společným jmenovatelem je podpora MQTT protokolu, jakožto možnosti komunikace zařízení s cloudovou službou.

V poslední části práce je navrhován požadovaný regulátor teploty tak, aby vyhověl všem požadavkům ze zadání. Práce se především zabývá návrhem DPS, jejím osazením a oživením. Součástí práce je výsledný firmware, který řídí celé zařízení. Oživením je proto chápáno pouhé otestování periférií, které jsou na desce osazeny a ověření jejich funkčnosti.

1. PROBLEMATIKA MĚŘENÍ TEPLOTNÍMI SENZORY

Elektrický senzor pro měření teploty je součástka, u které se mění její vlastnosti, většinou vodivost, v závislosti na okolní teplotě. V praxi se můžeme setkat i s jinými typy senzorů, například senzory pro bezdotykové měření teploty, které ale nejsou součástí této studie. Jejich použití pro tuto práci by postrádalo smysl, a navíc by bylo zbytečně dražší a složitější než použití dotykového čidla.

Dotyková čidla se dále dělí na elektrická, u kterých se mění fyzikální veličina, kterou lze v elektrotechnice snadno změřit, a na neelektrická. Typickým zástupcem neelektrických čidel je rtuťový teploměr, u kterého se mění objem a tím pádem je kapalina vytlačována vzhůru do kapiláry, na které je stupnice. Automatizované odečítání hodnot z takového neelektrického čidla by bylo velice náročné, nehledě na nutnost druhého čidla, které by do systému vneslo další chybu.

Dotykové elektrické senzory se dělí do dvou kategorií, a to podle toho, jakým způsobem je z nich vyčítána naměřená hodnota do komponenty, která signál dále zpracovává. První kategorií jsou analogové teplotní senzory, u nichž dochází ke změně elektrických vlastností v závislosti na teplotě. Druhou kategorií jsou digitální teplotní senzory. Ty jsou konstrukčně mnohem komplikovanější. Ve vnitřním zapojení takového digitálního senzoru se nachází analogový teplotní senzor, který je doplněn další periferií, která zpracovává naměřené hodnoty a ukládá aktuální hodnotu teploty do vnitřních registrů součástky.

1.1 Analogové teplotní senzory [1]

Jedná se o nejobecnější kategorii teplotních senzorů, proto je potřeba ji dále rozčlenit především podle použitých materiálů, ze kterých je dané čidlo vyrobeno. Základními skupinami jsou:

- kovová čidla,
- polovodičová čidla,
- termočlánky.

1.1.1 Kovová čidla

Jedná se o čidla vyrobená z velmi čistých kovů, například platina, nikl, molybden nebo měď. Uvnitř čidla se nalézá tenký drátek určité délky, který je stočený do cívky. Jako jádro této cívky jsou použity nevodivé materiály, jako například sklo nebo keramika. Typickým zástupcem této kategorie je teplotní čidlo Pt100, z jehož označení lze odvodit, že se jedná o platinové čidlo (Pt) a jeho odpor při 0 °C je 100 Ω.

Pro každý typ čidla existuje tabulka naměřených hodnot, která udává závislost odporu na teplotě. Odečítání teploty z kovového čidla probíhá tak, že je potřeba přesně změřit

odpor čidla, a pak pomocí tabulky lze odvodit hodnotu teploty, která náleží právě změřenému odporu.

Velkou nevýhodou těchto čidel je jejich křehkost a cena. Další nevýhodou je, že na čidlu dochází k relativně malé změně odporu při pokojových teplotách, proto je potřeba čidlo provozovat v zapojení s pomocným zesilovačem, což jeho použití opět prodražuje. Bezespornou výhodou čidla je jeho přesnost a velký pracovní rozsah, který je -50 až 400 °C. [2]

Při návrhu zesilovače pro Pt100 musí čidlem protékat proud menší než 1 mA, proto je třeba dbát zvýšené pozornosti při výběru odporů.



Obrázek 1.1 Teplotní senzor PT100

1.1.2 Polovodičová teplotní čidla

Základem každého polovodiče je velmi čistý monokrystal křemíku (Si), který je následně obohacen o příměsi prvků ze III. skupiny (B, Al, Ga, In). V takovém případě se jedná o polovodič typu P. Pokud se k monokrystalu přidá příměs prvků z V. skupiny (F, As, Sb), pak vytvoříme polovodič typu N. [1][3]

Polovodičové teplotní senzory se dělí na termistory a na monokrystalické odporové senzory. Termistory se dělí na pozistory (PTC) a na negastory (NTC).

Výroba termistorů je trochu rozdílná. Místo N a P přechodu se spíše využívají práškové oxidy kovů (Fe_2O_3 , TiO_2 , CuO , MnO , NiO , CoO , BaO), které se následně spékají. [4] Z toho vyplývá, že se nejedná o polovodiče v pravém slova smyslu, přesto se termistory řadí do polovodičových senzorů teploty, protože v nich nastává podobný děj jako v polovodičích. Se změnou teploty dochází k uvolnění nosičů elektrického proudu, tím pádem se materiál stává vodivým. [5]

PTC (*positive temperature coefficient*) termistory mají pozitivní teplotní součinitel odporu, zjednodušeně řečeno s rostoucí teplotou roste odpor. Jejich využití je hlavně v tom, že se dají použít jako vratné teplotní pojistky, například ve vinutí motoru. Měření teploty pomocí PTC termistoru není dobrou volbou, protože teplotní závislost je velice strmá. Častěji se používají pro kvantitativní měření, kde nejsou velké nároky na přesnost.

NTC (*negative temperature coefficient*) termistory mají negativní teplotní součinitel odporu, neboli s rostoucí teplotou klesá odpor termistoru. Jeho teplotní charakteristika se může aproximovat exponenciálou, proto odečítání umožňuje měřit malé změny teploty bez větších potíží. Malou nevýhodou je, že každá součástka vykazuje trochu jiný průběh teplotní charakteristiky, který se může časem mírně měnit. Výhodou NTC termistoru je jeho cena. Díky tomu se často používá pro bodové měření teploty. [4]



Obrázek 1.2 NTC termistor 640-10K



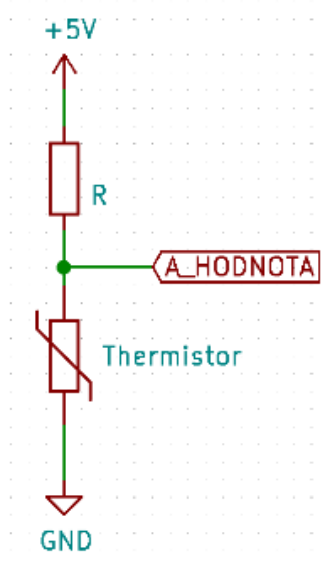
Obrázek 1.3 NTC termistor B57164K0472K000

Monokrystalické senzory se vyrábějí z křemíku (Si), přesněji pomocí polovodiče typu N. S rostoucí teplotou u nich dochází k rozptýlu nosičů. To má za následek snížení vodivosti neboli růst odporu. Oproti PTC termistorům mají mírnější strmost teplotní charakteristiky, proto jsou vhodné pro měření teplot v rozsahu od -55 do 150 °C. Typickým zástupcem této skupiny je teplotní senzor s označením KTY. [5]



Obrázek 1.4 Senzor KTY81 110

Nejjednodušším způsobem, jak můžeme analogové čidlo připojit k mikrokontroleru, je zapojit ho do série s rezistorem a pomocí analogového převodníku vyčítat úbytek napětí na čidle, viz Obrázek 1.5. Jedná se o opravdu jednoduché zapojení, kde nemůžeme klást velké nároky na přesnost. Přesně lze měřit pouze teploty blízké pokojové teplotě. Navíc vyčítaná hodnota není lineární. To je způsobeno použitím AD převodníku. Přesnost měření obzvláště ovlivňuje počet jeho kvantizačních úrovní.



Obrázek 1.5 Jednoduchý způsob odečítání hodnot z analogového teplotního čidla

1.2 Digitální teplotní senzory

Digitálními teplotními senzory se chápou součástky, které v sobě mají implementováno měření teploty pomocí analogového teplotního senzoru. Takto naměřená teplota je následně digitálně zpracována a uložena do registru uvnitř součástky. Pro získání dat musí být senzor připojen na sběrnici. Poté je možné se součástkou komunikovat a vyčítat z ní data nebo nastavit v jakých jednotkách (°C nebo °F) má data posílat. Nejčastějšími typy sběrnic/komunikačních protokolů jsou I2C, 1-Wire, SPI. Typickými zástupci v této kategorii jsou DS18B20 nebo AM2302.

1.2.1 DS18B20 [6]

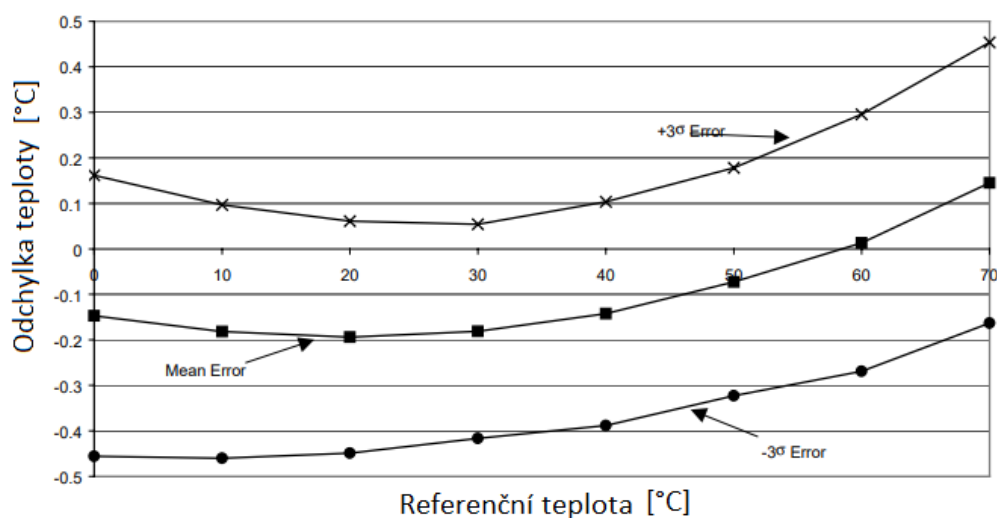
Digitální teplotní senzor měří teplotu v rozsahu od -55 do 125 °C. Poprvé ho vyrobila firma DALLAS SEMICONDUCTOR, proto se mu lidově říká „dallas“. Senzor komunikuje pomocí 1-Wire sběrnice. Teplota je v senzoru uložena jako 12 bitové slovo.

Další možností, které čidlo nabízí, je nastavení přesnosti měření, a to v rozsahu 9 až 12 bitů. Snížení přesnosti měření urychluje měřicí proces. Díky tomu je možné několikanásobně urychlit měření, které je při nejvyšší přesnosti velice pomalé.

Čidlo je dále vybaveno interní logikou, díky které se samo může chovat jako termostat.



Obrázek 1.6 Teplotní senzor DS18B20 [6]

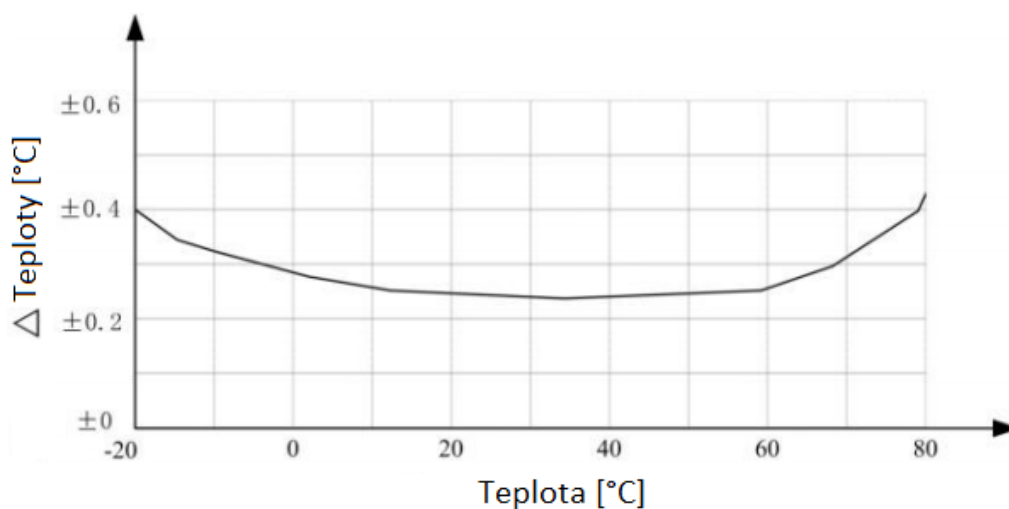


Obrázek 1.7 Graf odchylky při měření teploty pomocí DS18B20

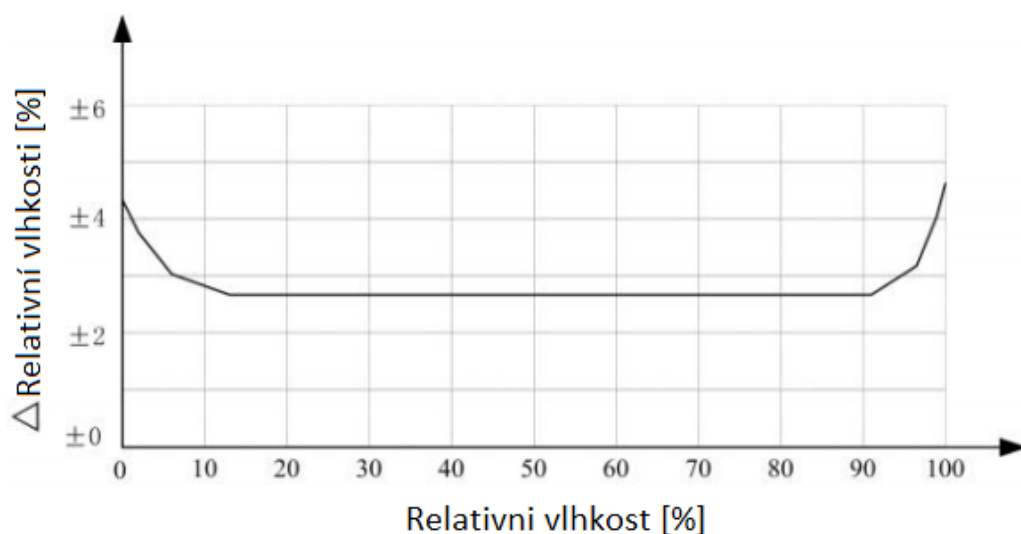
1.2.2 AM2302 [7]

Jedná se o další často využívané čidlo především proto, že je součástí DHT22 teplotního senzoru. Velkou výhodou čidla je, že dokáže s teplotou měřit i relativní vlhkost vzduchu. Jeho nevýhoda je nepřesnost (viz Obrázek 1.8) oproti předchozímu DS18B20 (viz Obrázek 1.7), přestože AM2302 je schopno rozlišit měřenou hodnotu teploty na 16 bitů.

Pro komunikaci čidlo využívá Single-bus komunikační protokol. Jedná se o upravenou verzi 1-Wire protokolu, přesto je nelze zaměňovat.



Obrázek 1.8 Graf odchylky při měření teploty pomocí AM2302 [7]



Obrázek 1.9 Graf odchylky při relativní vlhkosti pomocí AM2302 [7]

1.3 Termočlánky [1]

Termočlánek využívá termoelektrického jevu. U tohoto jevu vzniká na rozhraní dvou různých kovů malé napětí, které závisí na teplotě (Seebeckův jev). Díky tomu lze termočlánek použít pro měření velkého rozsahu teplot s relativně velkou přesností. Bezespornou výhodou tohoto typu teplotního sensoru je jeho časová stálost.

Termočlánky se dělí podle kombinace kovů anebo podle jejich kalibrace. Nejčastěji se můžeme setkat s kalibracemi, které se značí J, K, E, T (viz obrázek 1.8, kde je přehled jejich vlastností).

Označení termočlánku dle IEC 584	Původní označení	Měřicí rozsah [°C]
T	Cu-CuNi, Cu-ko	- 200 až 350
J	Fe-CuNi	- 200 až 750
E	NiCr-CuNi, ch-ko	- 100 až 900
K	Ni-Cr-Ni, ch-a	- 200 až 1200
N	NiCrSi-NiSi	- 200 až 1200
S	PtRh10-Pt	0 až 1600
R	PtRh13-Pt	0 až 1600
B	PtRh30-PtRh6	300 až 1700

Obrázek 1.10 Tabulka přehledu termočlánků [8]

2. ROZBOR IOT CLOUDŮ

Cloudové služby pro IoT jsou v dnešní době rychle se rozvíjející služby, jejichž cílem je logování a vizualizace dat. Každá služba je specifická. Především se liší v tom, jak je možné data zobrazovat, kolik čidel lze připojit, jak často můžou být data do cloudu posílána nebo specifikací, kde čidlo musí být umístěno.

Důležitým faktorem při výběru IoT cloudu je především to, pro jaké zařízení nebo pro jaký typ čidel je služba určena. Některé služby jsou vyhrazeny pouze pro originální Arduino zařízení. Existuje služba, která je určena především pro venkovní senzory. Kupodivu se v tomhle případě nejedná o nějaký nesmyslný zákaz, ale služba se snaží chránit samotné uživatele.

Společným průsečíkem většiny služeb je podpora MQTT protokolu, který se využívá pro odesílání dat do cloudové služby. Tento protokol je vysvětlen dále v práci.

V této době existuje velké množství těchto služeb, seznam nejpoužívanějších služeb můžeme nalézt v Tabulka 2.1.

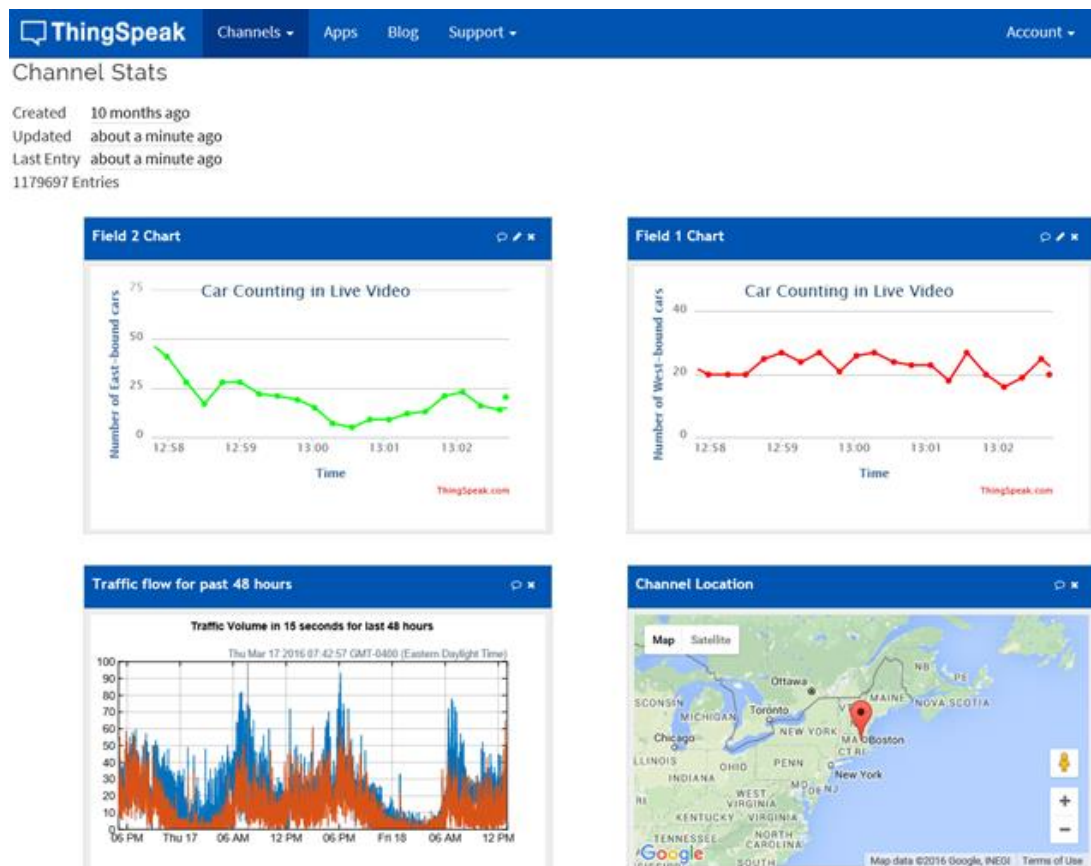
Tabulka 2.1 Tabulka nejpoužívanějších IoT cloudů

Název cloudu	Webová stránka projektu
ThingSpeak	https://thingspeak.com/
TMEP	https://tmep.cz/
IoT GURU cloud	https://iotguru.live/
ArduinoIoTcloud	https://login.arduino.cc/
openHAB	https://www.openhab.org/
IBMCloud	https://cloud.ibm.com/login
Cloud IoT Core	https://cloud.google.com/iot-core
Fogwing	https://www.fogwing.io/
Axonize	https://www.axonize.com/platform/
Blynk	https://blynk.io/
DATADOG	https://www.datadoghq.com/product/
Bevywise	https://www.bevywise.com/iot-platform/

2.1 ThingSpeak

Jedná se o online aplikaci, která je v poslední době velice rozšířená. K jejím přednostem patří především to, že ji lze využívat bezplatně, ovšem s určitými omezeními. Bezplatná verze nabízí správu maximálně 4 kanálů s celkovým maximálním počtem 3 milionů zpráv za rok. Dalším kritériem je limit intervalu aktualizace zpráv, a to 15 sekund. Data uložená v cloudu jsou omezena u bezplatných účtů na 10 milionů. Z toho vyplývá, že data v cloudu budou uložena přes 3 roky s minimálním intervalem aktualizace. [9].

Důvodem, proč je tato služba v poslední době velice rozšířená je, že kolem aplikace existuje velká komunita lidí, která vydává mnoho návodů, jak to používat. Velkou výhodou je možnost zpracovávat přijatá data pomocí matlabu, díky kterému můžeme data přepočítávat, vyvozovat z nich závěry a zobrazovat nejrůznější průběhy v grafech.



Obrázek 2.1 Ukázka z uživatelského rozhraní ThingSpeak [13]

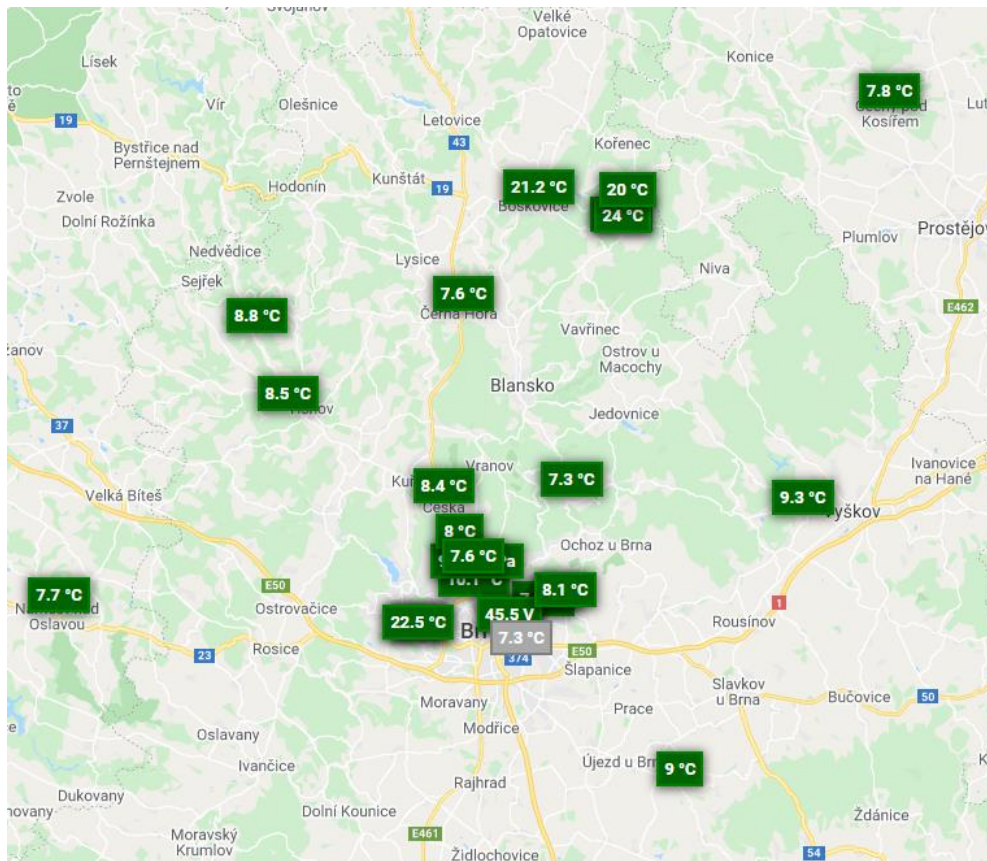
2.2 TMEP [10]

Cloudová služba TMEP je ryze českou záležitostí. Zaměřuje se pouze na venkovní čidla na území České republiky a v blízkém okolí. Motivaací této služby je shromažďovat meteorologická data z celého území ČR, proto je potřeba sdílet i GPS souřadnice čidla. Většina uživatelů vystavuje svoje čidla veřejně tak, aby se k jejich naměřeným datům mohl kdokoli dostat.

Provozování tří čidel bez komerčního užití je zcela zdarma, dokonce údaje z čidel nemusí být veřejně vystaveny. Stačí, když u čidla nejsou uvedeny GPS souřadnice. Podmínkou pro zajištění správné činnosti je potřeba posílat data každou minutu, v nejhorším případě každých pět minut. Zásadní informací je, že služba nepodporuje MQTT protokol. Místo toho jsou data odesílána pomocí HTTP GET dotazu na port 80.

Teplota není jediný parametr, který lze posílat. Často se teplota kombinuje s vlhkostí vzduchu nebo atmosférickým tlakem. Výjimkou není ani zasílání hodnoty napětí na baterii.

Další výhodou je, že zjednodušená verze celého projektu podporující jedno čidlo je vystavena jako open source a dá se stáhnout z portálu GitHub.



Obrázek 2.2 Mapa čidel vyexportovaná 6.12.2020 v 17:15 [10]

2.3 openHAB

Jak už je patrné z názvu, jedná se o open source projekt, který je primárně určený k automatizaci chytrých domů. Cloudová služba předpokládá provoz vlastního serveru v lokální síti nebo serveru ve veřejném internetu, která bude jenom pro účely daného chytrého domu. S vidinou připojení vyvíjeného termostatu by taková služba byla hodně velký a finančně nákladný luxus. Ovšem komunita z openHAB foundation provozuje i online službu myopenHAB, která je zdarma. [11]

MyopenHAB je sice především určen k prezentačním a testovacím účelům, přesto nezakazuje využití této aplikace pro provoz osobních čidel. Ale má to malé riziko. Služba negarantuje to, že bude stále fungovat nebo že se data po čase nesmažou. Použití je na vlastním uvážení.

Tato služba je pro účely vyvíjeného termostatu nebo na testování během vývoje optimální. Hlavně pokud je cílem používat termostat spíše nárazově, jednou za čas. Poslední dobou není problém použít malý deskový počítač např. raspberryPi, který lze v domácnosti používat jako server, na kterém by mohla běžet tato aplikace. V dnešní době je mnoho domů, které už takový malý domácí server mají a v budoucnu se zvyšujícím se zájmem o inteligentní domy bude takový server standard, proto by bylo dobré, aby byl termostat kompatibilní a snadno připojitelný i na tento cloudový systém. Podpora MQTT protokolu je u této aplikace samozřejmostí.

2.4 MQTT

Protokol lze najít pod označením ISO/IEC 20922 a jeho zkratka znamená „*Message Queuing Telemetry Transport*“, v překladu: „*zprávu dotazující telemetrický přenos*“. MQTT pracuje na sedmé aplikační vrstvě a využívá TCP/IP protokolu. Je navržen tak, aby byl lehký a moc nezatěžoval datový přenos. Hlavním cílem je publikování a přijímání zpráv mezi zařízeními. Vyznačuje se obousměrností, bezztrátovostí a neprohazováním pořadím zpráv. Díky tomu se skvěle hodí pro komunikaci mezi zařízeními stroj-stroj nebo k nasazení do IoT zařízení, protože nezatěžuje běh zařízení a nepotřebuje nebo nemá velkou šířku pásma, dokonce je odolný proti velkým latencím v síti.

V dnešní době existuje mnoho verzí MQTT, nejaktuálnější je MQTT version 5.0. I přesto je potřeba dodat, že protokol neřeší žádnou autentizaci a bezpečnost, proto je dobré ho zkombinovat ještě s TLS. Protokol dále podporuje QoS, díky kterému je možnost zprávě přiřadit prioritu podle toho, jak je důležitá. Priorita zpráv se dělí do tří kategorií: doručit maximálně jednou, doručit alespoň jednou a doručit pouze jednou. [12]

3. HARDWARE

V zadání bakalářské práce je požadavek na použití ESP8266, což je mikrokontroler, který v sobě obsahuje rozhraní pro komunikaci pomocí wifi standardů. Pomocí tohoto rozhraní bude zařízení schopno odesílat data do IoT cloudu anebo komunikovat s uživatelem.

Termostat by neplnil svoji funkci, kdyby neuměl měřit teplotu, zařízení proto obsahuje konektor, do kterého lze připojit čidlo DS18B20 nebo DHT22. Pro připojení čidla byl vybrán konektor RJ13 (samice 4P4C), do kterého se zasunuje konektor RJ10 (samec 4P4C).

Ústředním funkčním blokem celého zařízení je část obvodu ovládající výstupní zásuvku. Pro toto zařízení bylo zvoleno spínání zásuvky pomocí relé, které bude spínat přívod elektrického odporového topení. S ohledem na to, že zařízení bude připojeno do běžné zásuvky, která je nejčastěji dimenzována a jistěna na jmenovitý proud 16 A, bylo zvoleno relé, která má stejný jmenovitý proud. Důsledkem toho byl stanoven maximální spínaný výkon na 3,6 kW.

Důležitým blokem toho zařízení je externí I2C EEPROM, na kterou jsou ukládány aktuální nastavení a konfigurace, především informace o wifi sítích, ke kterým se má zařízení připojovat. Nastavení síťového rozhraní v režimu AP a klíče pro odesílání dat na IoT cloud.

Dalším funkčním blokem je převodník USB to UART, pomocí kterého je možné zařízení naprogramovat a ladit při vývoji firmwaru. Jedná se obvod s převodníkem CH340G a s pomocnými součástkami, díky kterým celý blok správně pracuje.

Součástí práce není návrh zdroje, přesto bylo potřeba vybrat vhodný zdroj, kterým by bylo celé zařízení napájeno. Požadavky na zdroj byly následující: vstupní napětí 230 V AC, výstupní napětí 5 V DC a výstupní proud kolem 1 A. První volbou byl malý deskový zdroj, který se na první pohled tvářil bezpečně a spolehlivě. Po zevrubnějším prozkoumání se ukázalo, že ve zdroji nejsou dodrženy bezpečné odstupy cest se síťovým vypětím. Později se ukázalo, že zařízení značně ruší své okolí, toto ručení pravděpodobně zasahovalo i do pásma 2,4 GHz, protože došlo ke značnému poklesu přijímané úrovně signálu. Z toho důvodu bylo potřeba tento zdroj nahradit jiným.

Při návrhu desky plošných spojů se objevilo mnoho problémů, to vedlo k několika verzím DPS, ve kterých byly postupně odstraňovány chyby. V posledních verzích došlo k rozdělení desky na dvě části, zdrojovou a výpočetní. Tento koncept se zachoval i v konečné verzi. Hlavní motivací k tak radikální změně byla implementace bezpečného, zato rozměrově většího zdroje MYRRA 47162. Rozdělení na dvě části přineslo i zvýšení bezpečnosti zařízení, protože umožnila dostat do zařízení druhé relé, díky kterému zařízení spíná všechny pracovní vodiče. Díky tomu máme jistotu, že na spínané zásuvce se ve vypnutém stavu neobjeví fázové napětí.

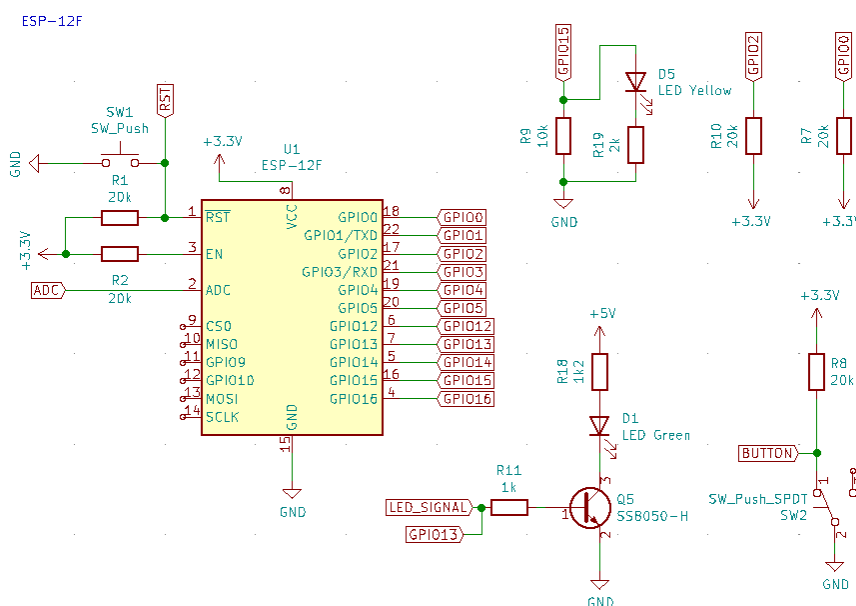
3.1 Mikrokontrolér a jeho periferie

Návrh tohoto bloku je především zaměřen na periferie modulu ESP8266. Hlavním úkolem této části je zajistit potřebné napěťové úrovně na pinech GPIO15, GPIO0 a GPIO2, aby modul mohl nabootovat z interní SPI flash paměti nebo provést nahrání nového firmwaru, viz tabulka 3.1. Připojení pinu na určitou logickou úroveň H, která značí 3.3 V a na logickou úroveň L, která se rovná 0 V, musí být provedeno pomocí odporů, které omezují proud.

Mezi další komponenty patří resetovací tlačítko, které při stisknutí přemostí rezistor R1 a přivede log 0 na pin RST, čímž resetuje ESP zařízení. Dále jsou k MCU připojeny dvě kontrolní LED diody. První LED dioda zelené barvy je spínaná pomocí tranzistoru a druhá žlutá je připojená přímo na GPIO15 pin, který je možné po startu zařízení použít jako výstupní pin. Poslední součástka, kterou zbývá popsat je přepínač, pomocí kterého je možné přepínat režimy zařízení. Popiska BUTTON dále je ve schématu připojena k GPIO16. V předchozích verzích DPS byla k přepínači připojena kontrolní LED, která indikovala, v jakém režimu se zařízení nachází. Od této myšlenky bylo upuštěno, protože byla potřeba ovládat tuto diodu softwarově, aby bylo možno rozpoznávat více stavů než dva. Díky této změně teď lze rozpoznávat stavy vypnuto – blikající - zapnuto.

Tabulka 3.1 Tabulka potřebných napěťových úrovní pro ESP8266

GPIO15	GPIO0	GPIO2	Mód
L	H	H	Boot z SPI flash paměti
L	L	H	Programování pomocí UART



Obrázek 3.1 Blokové schéma zapojení MCU a jeho periférií

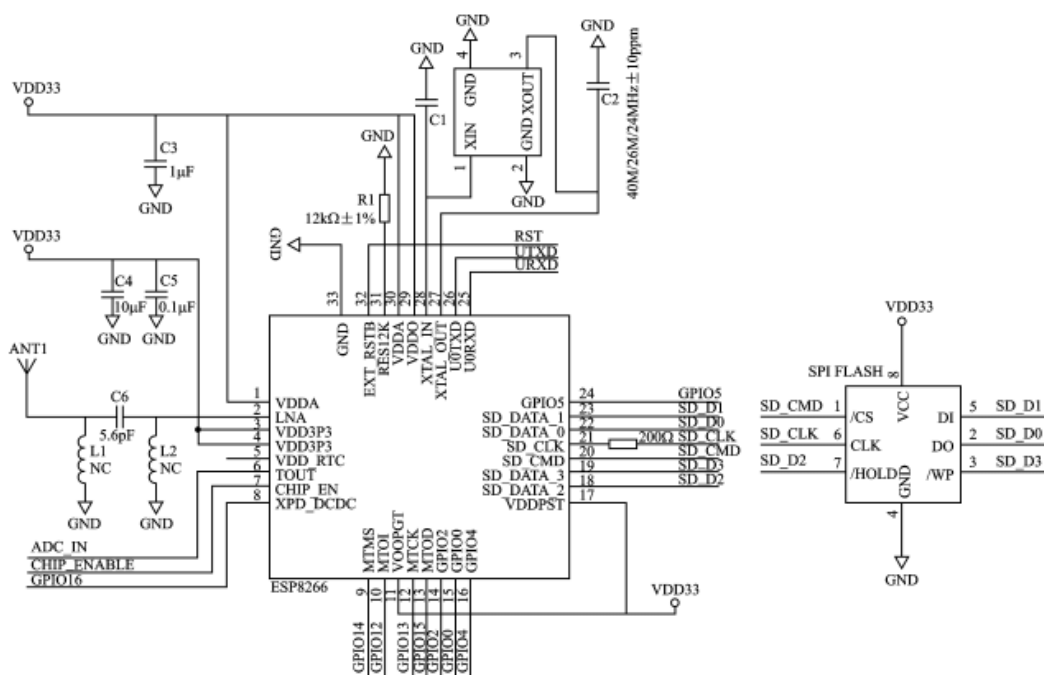
3.1.1 ESP8266

Jedná se o relativně malou destičku, kterou je možné použít místo mikrokontroleru. V dnešní době existuje mnoho variant a provedení od ESP-01 po ESP-12. Rozdíl mezi jednotlivými typy je především v počtu vyvedených pinů a velikosti interní EEPROM paměti, do které se ukládá program. Dalšími parametry jsou rozměry modulu, uspořádání pinů, tvar a provedení antény.

Provedení ESP-12 se dále dělí na ESP-12E a ESP-12F. Jediným rozdílem v těchto verzích je velikost a tvar antény. Modul verze 12F má parametrově lepší integrovanou anténu, z toho důvodu byl pro daný projekt zvolen právě tento modul.

Dalším důvodem je velká EEPROM paměť, která má velikost 4 MB, dokonce jsou na pouzdro modulu vyvedeny piny SPI sběrnice, na kterou je připojena výše zmíněná interní EEPROM paměť. V neposlední řadě je zpětně kompatibilní s ESP-07. ESP-07 má jednu velkou výhodu, a to je možnost připojení externí antény. To v projektu umožní, že v případě malého pokrytí lze pouze vyměnit desku ESP modulu a přidat externí anténu bez nutnosti návrhu celé nové DPS.

ESP obecně má velmi malé nároky na spotřebu proudu, k běhu potřebuje pouze 80 mA a díky tomu je vhodný pro nasazení do aplikací, kde je použito napájení z baterií. Nevýhodou je spotřeba při startu, při kterém může ESP potřebovat až 500 mA. Pokud by při startu nebyl dodán dostatečný proud, který čip potřebuje, může se stát, že nenaběhne správně. Mohlo by se pak stát, že se bude chovat náhodně nebo nebudou fungovat některé periferie. Z tohoto důvodu je v projektu použit dostatečně výkonný zdroj, který je schopen dodávat až 700 mA dlouhodobě s krátkodobým zatížením až na 1 A.

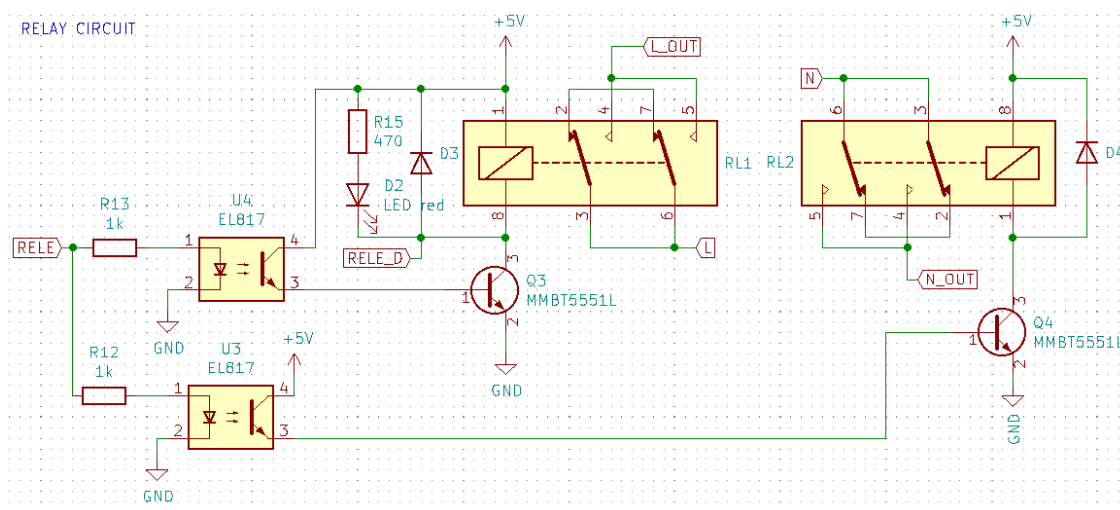


Obrázek 3.2 Schéma desky modulu ESP [18]

V každém modulu je osazen čip s označením ESP8266EX. Prvotním záměrem výrobce bylo vyrobit komunikační modul, který by bylo možné ovládat AT příkazy a pro přenos dat by využíval již zavedený IEEE 802.11 protokol. Jedná se o plnohodnotný mikrokontroler s architekturou Tensilica L106 32-bit RISC, díky které je velice nenáročný na spotřebu energie.[14] Navíc je v čipu rovnou implementováno celé wifi rozhraní. Dostatečný výkon tohoto mikrokontroleru umožňuje vykonávat jednoduchý firmware.

3.2 Ovládání spínané zásuvky

V další části jsem se zaměřil na periférii zařízení, která má být schopná spínat topení nebo výhřevný spotřebič. Existuje mnoho způsobů, kterými by šel tento problém vyřešit. Za jedno z možných řešení lze považovat regulaci pomocí triaku nebo párem tyristorů. Elegantnější a výrazně spolehlivější způsob je použití relé pro spínání všech pracovních vodičů. Tento typ regulace sice není spojitý, pracuje jen ve stavech vypnuto/zapnuto, zato je zde galvanicky odděleno ovládací napětí od spínaných proudů. Při výběru by se nemělo zapomenout na to, že relé musí být schopné spínat síťové napětí 230 V a proud 16 A. Toho bylo docíleno použitím relé s dvojitým přepínačem. Dvojice přepínačů v jednom relé byla paralelně spojena, tím byl dvojnásobně navýšen jmenovitý spínaný proud.



Obrázek 3.3 Schéma zapojení obvodu pro spínání výstupní zásuvky

Na obrázku 3.3 je vidět zapojení obvodu pro spínání dvou relé. Každé relé je připojeno na zdroj 5 V a spínáno tranzistorem NPN MMBT5551L. Každý z těchto tranzistorů je spínán přes optočlen EL817, který je zde použit jen jako galvanický oddělovač. Optočleny jsou ovládány vstupním signálem označeným jako RELE, ten je připojen přímo na pin ESP modulu. Dalšími součástkami, které jsou v obvodu použity jsou ochranné diody, díky kterým při sepnutí/rozepnutí relátka dochází k demagnetizaci

cívky relé a tím chrání spínací tranzistory proti napětovým špičkám. Další důležitý prvek je LED dioda, která indikuje, jestli je relé sepnuto. Při návrhu DPS byla tato indikační LED dioda přesunuta ze zdrojové části do výpočetní, a to z důvodu, aby byly všechny kontrolní LED diody na jednom místě. Přesto ve schématu zůstala umístěna zde, a to hlavně kvůli tomu, aby bylo jasné vidět, jakým způsobem je stav LED diody vyčítán.

Bezpečnost tohoto bloku je nejvyšší prioritou, proto v klidovém stavu (bez úrovně 3,3 V na signálu RELE) jsou relátka vypnuta. Díky tomu se v případě poruchy na výstupní zásuvce neobjeví síťové napětí a dojde k odstavení ovládaného tepelného spotřebiče. Dalším bezpečnostním prvkem je použití dvou relátek pro spínání všech pracovních vodičů (díky tomu vypneme vždy i fázový vodič).

Volba tranzistoru byla provedena následovně: napětí napájecího zdroje U_{VCC} je 5 V, proto je potřeba hledat tranzistor, který snese alespoň trojnásobek tohoto napětí, tedy 15 V, což splňuje skoro každý tranzistor. Dalším důležitým parametrem je maximální proud spínaný tranzistorem. Ten je vypočten v rovnici 3.1. $R_{cívky}$ je katalogová hodnota pro zvolené relé a značí odpor cívky.

$$I_{c, tranz} = \frac{U_{VCC}}{R_{cívky}} = \frac{5}{62} = 81 \text{ mA} \quad (3.1)$$

Vypočtená hodnota je 81 mA, ale stejně jako u pracovního napětí je potřeba tuto hodnotu ztrojnásobit. Je hledán tranzistor s hodnotou maximálního proudu alespoň 250 mA. Proud procházející tranzistorem může být chvilkově vyšší než teoreticky spočtený, například při zapnutí nebo vypnutí.

Výpočet potřebného proudu pro sepnutí relé je uveden v rovnici 3.2, kde β_{tranz} je zesílení tranzistoru a U_{VCC} je 5 V. Poslední hodnota $R_{cívky}$ je katalogová hodnota pro zvolené relé a značí odpor cívky.

$$I_{b, tranz} = \frac{I_{c, tranz}}{\beta_{tranz}} = \frac{U_{VCC}}{\beta_{tranz} \cdot R_{cívky}} = \frac{5}{80 \cdot 62} = 1 \text{ mA} \quad (3.2)$$

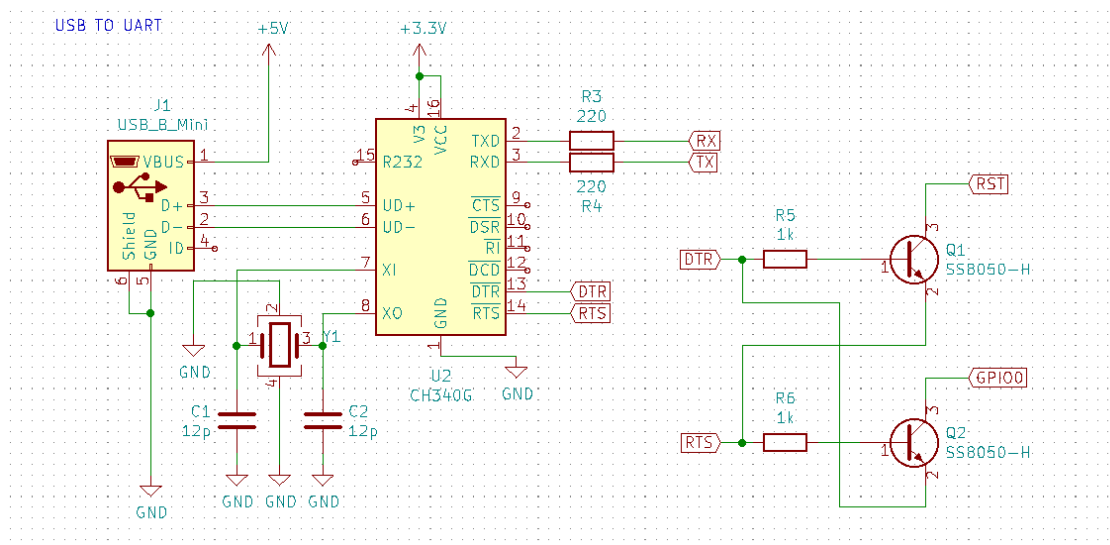
Z předchozího výpočtu (rovnice 3.2) jasně plyne, že tranzistor je možné spínat přímo pinem na mikrokontroleru, kde je pro ESP8266 povolen proud 12 mA. Přesto je zde ještě začleněn optočlen, proto je hodnota rezistorů R12 a R13 určena experimentálně na 1 k Ω .

Hodnota odporu R15 u LED diody je vypočtena pomocí rovnice 3.3, kde U_{VCC} je napájecí napětí 5 V, U_d a I_d je pracovní bod diody.

$$R_{15} = \frac{U_{VCC} - U_d}{I_d} = \frac{5 - 1,7}{0,007} = 470 \text{ } \Omega \quad (3.3)$$

3.3 Převodník z USB na UART

Další důležitou částí návrhu je převodník z USB na UART, viz Obrázek 3.4. Jeho hlavním prvkem je integrovaný obvod CH340G, který se využívá i v mnoha jiných vývojových deskách s ESP [15]. Jeho výhodou je to, že z jeho pouzdra jsou vyvedeny i ovládací piny pro RS232. Dva z těchto pinů se využívají jako řídicí signály pro přepnutí ESP do režimu bootu nebo běhu. Díky tomu je možné mikrokontroler programovat bez nutnosti nějaké jiné akce na desce, jako například přepnutí tlačítka nebo přidání propojky.



Obrázek 3.4 Schéma zapojení převodníku z USB na UART

Důležitou součástí tohoto schématu je klopný obvod, který má na starosti ovládat logickou hodnotu na GPIO0 pinu tak, aby v klidové stavu byl tento pin připojen na Log. 1 a v případě nahrávání nového firmwaru vyresetovat MCU a nastavit na pin GPIO0 Log. 0. viz tabulka 3.1.

Integrovaný obvod CH340G nemůže fungovat bez externího 12 MHz krystalu, který celý systém taktuje. Pro napájení IO je použito napětí 3,3 V a to hlavně kvůli kompatibilitě UART protokolu s ESP modulem tak, aby oba pracovaly na stejné napěťové hladině. Pokud by to tak nebylo, musel by se použít pomocný obvod, který by převáděl 5 V UART na 3,3 V a obráceně. K tomuto účelu by se dala použít Zenerova dioda pro snížení napěťové úrovně.

Mini USB zde bylo použito hlavně kvůli jeho robustnosti, a protože se snadněji osazuje na DPS než USB Micro. Jeho nevýhodou je velikost, na desce zabírá poměrně hodně místa. Tato nevýhoda se projevila v prvních návrzích desek, ovšem od verze s více DPS není tato nevýhoda tak zásadní.

3.4 I2C EEPROM paměť

Další součástí, kterou obvod obsahuje je EEPROM paměť. V původním návrhu byla použita M24512, která má 512 kbit. V konečné verzi byla nahrazena pamětí 24LC16B s 8x255 bajtů, a protože obě paměti fungují koncepčně jinak, bylo potřeba drobných úprav.

Obě paměti jsou připojeny pomocí I2C sběrnice. Paměť je zamýšlena jako úložiště konfiguračních dat, jako například jméno a heslo na wifi síť.

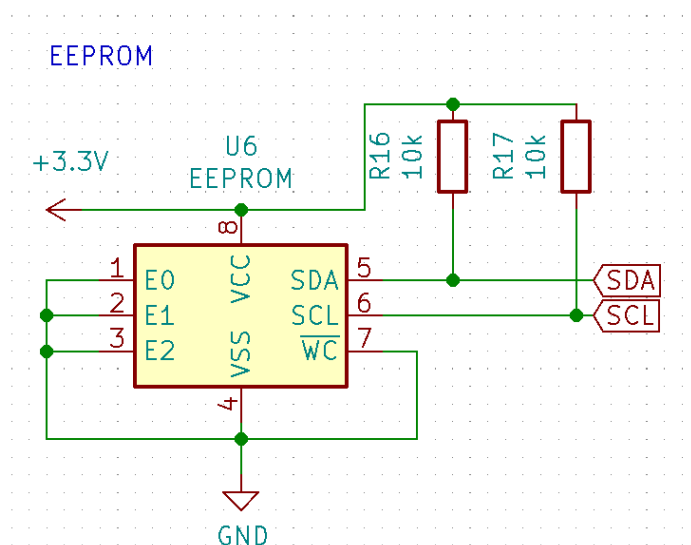
Na obrázku 3.5 je vidět piny E0, E1 a E2. Těmito piny je možné u běžné paměti přesněji nastavit poslední 3 bity v adrese zařízení. V bakalářské práci jsou všechny tyto piny nastaveny na hodnotu logické 0, neboli jsou připojeny na GND. Navíc u paměti 24LC16B jsou takto nastavené hodnoty ignorovány, protože je členěna na osm bloků po 255 bajtech. Výběr bloku je realizován v adrese. Z toho vyplývá, že EEPROM se tváří jako osm EEPROM zařízení na jedné sběrnici, a to bez ohledu na to, jak jsou zapojeny piny E0-2. [19]

3.4.1 I2C

Jedná se o datovou sběrnici využívající pro obousměrný přenos dat pouze dva vodiče, SDA a SCL. Sběrnice byla vyvinuta firmou Philips v roce 1982 [20], dnes pod firmou NXP Semiconductors. Jedná se o chráněnou značku, proto se lze setkat i s označením TWI, což je téměř identická sběrnice jako I2C.

Vodič s označením SCL přenáší hodinový signál s frekvencí 100 kHz nebo 400 kHz a vodič SDA přenáší data. Na sběrnici je v ideálním případě možné umístit až 128 zařízení. Pro přenos je potřeba stanovit, které zařízení je master a které slave.

Pro zajištění funkčnosti sběrnice je potřeba připojit oba vodiče SCL a SDA pomocí rezistorů R16 a R17 k napájení, viz obrázek 3.5.



Obrázek 3.5 Schéma zapojení EEPROM paměti

3.5 Výroba a návrh DPS

Před samotným návrhem DPS je potřeba vybrat krabičku. Pro tento projekt byla vybrána krabička od firmy KRADEX s výrobním označením Z27J, která už obsahuje dva konektory, jeden typu samec, druhý typu samice, oba pro síťové napětí 230 V, které se používají na území České republiky.

Po výběru krabičky bylo přikročeno k návrhu samotné desky. Rozměry desky byly přímo přizpůsobeny tak, aby se vlezla do vybrané krabičky a taky, aby seděly montážní otvory na předpřipravené plošky.

Návrh desky a kreslení schémat bylo provedeno v programu KiCAD, který byl zvolen hlavně kvůli tomu, že umožňuje exportování 3D modelu navržené desky. Další výhodou programu je velká knihovna součástek a jejich snadná editace. Celý program je distribuován jako open source. Díky tomu existuje velká komunita lidí, kteří jsou ochotni poradit s jakýmkoli problémem, který je potřeba v programu vyřešit.

Návrh počítá s použitím SMD a vývodových součástek, a to hlavně kvůli tomu, aby se ušetřilo místo na desce, rozměry desky jsou limitovány rozměrem krabičky. U některých prvků, především u relé, přepínače, zdrojové desky a konektorů není možné použít povrchovou montáž. U signalizačních LED diod je také uvažována vývodová montáž a to z důvodu, aby se LED diody mohly namontovat do krytu krabičky a nemusely být řešeny světlovody, které by světlo přenášely z desky na povrch krabičky.

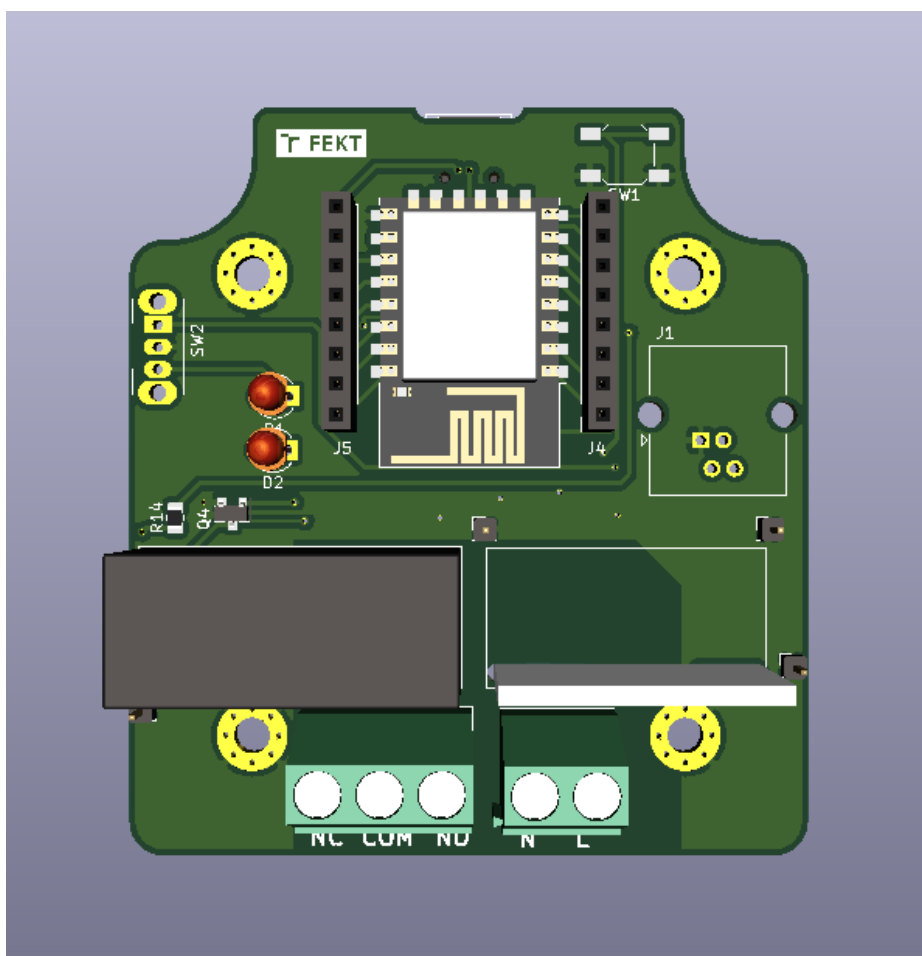
Do projektu také byla přidána patice na připojení dalších prvků. Pro tuto patici byla zvolena rozteč a uspořádání pinů jako na desce Wemos [15]. Díky tomu při vývoji není nutné osazovat desku přímo ESP modulem, ale může se využít toho, že se na tuto patici připojí deska Wemos. Z důvodu rozdělení zařízení na dvě desky už není možné namontovat výpočetní desku s Wemos zařízením, ale je potřeba DPS osadit ESP modulem, jinak se zařízení nevleze do krabičky.

Postupný vývoj zařízení vyústil ve tři rozdílné verze zařízení, první verze (černá) je navržena pouze na jedné desce. Důvodem k opuštění této verze byla nevhodná volba zdroje, který nevyhovoval evropským standardům. Nesplňoval bezpečnostní odstupy síťového napětí na PDS zdroje, navíc zdroj podstatným způsobem rušil ESP, což vedlo ke snížení citlivosti přijímaného wifi signálu. Další verze byla doplněna o MYRRA zdroj, kvůli kterému bylo potřeba zařízení rozdělit na dvě DPS. Díky tomu v zařízení vznikl dostatek místa na to, aby se do něj mohlo umístit ještě jedno relé. Tato verze byla mnohonásobně bezpečnější než první verze. Protože při návrhu druhé verze jsem zapomněl nakreslit jednu cestu, musel jsem přistoupit k návrhu třetí verze, která obsahuje pouze novou výpočetní DPS. Spodní zdrojová DPS byla zachována z verze 2. Pro snadnější montáž a vycentrování desky byl potřeba navrhnout speciální distanční blok, díky kterému je možné desky spojit a namontovat do krabičky. Tento distanční blok byl

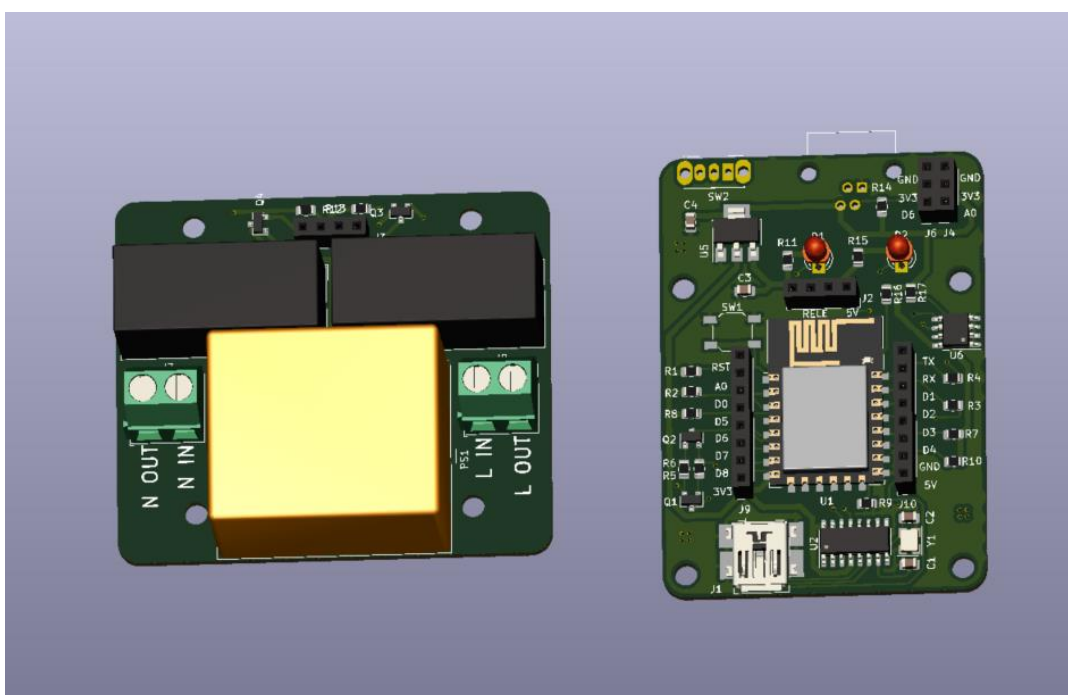
navržen v programu SOLIDWORKS a následně vytisknut na 3D tiskárně z transparentního PLA s výškou vrstvy 0,2 mm.

Na následujících obrázcích 3.6 až 3.10 jsou zobrazeny 3D modely, které byly vytvořeny pomocí programu KiCad. Jedná se o modely jednotlivých verzí desek, které byly v rámci práce zpracovány.

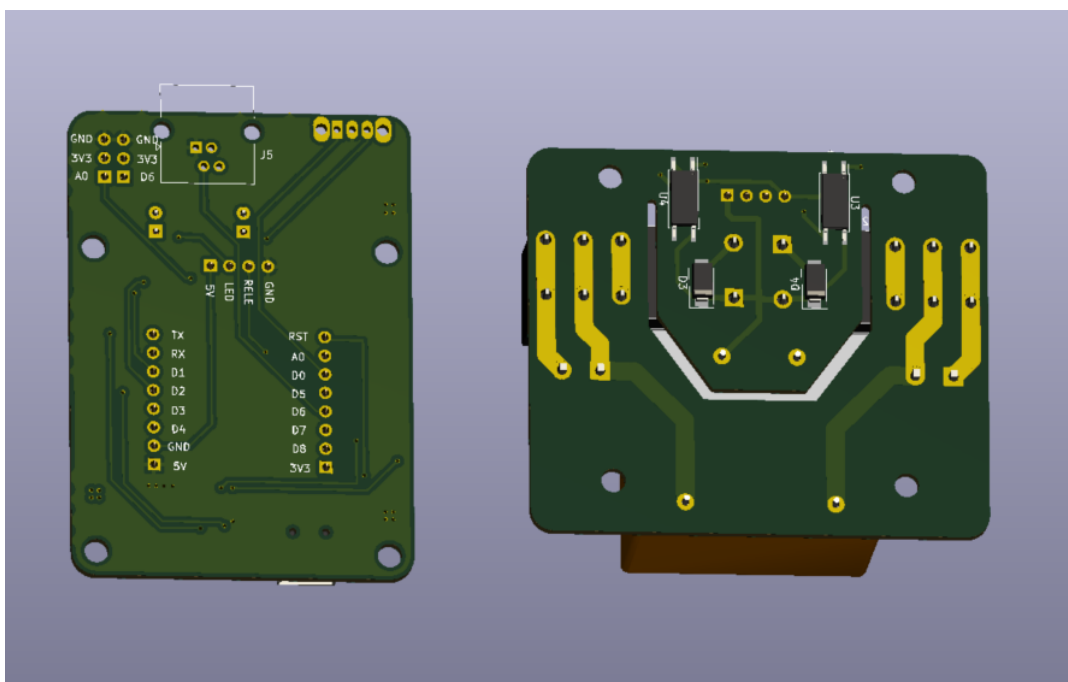
Všechny DPS byly vyrobeny zahraniční firmou, výběr této společnosti byl závislý hlavně na výsledné ceně. Snahou bylo vybrat takovou společnost, u které bude výroba desek co nejnížší. Značnou nevýhodou výroby desek v zahraničí je její časová náročnost, která je z velké části způsobena dobou přepravy, která je několikanásobně vyšší než potřebný čas na výrobu. Většina výrobců požaduje podklady pro výrobu ve formátu GERBER. Jedná se o otevřený vektorový formát ASCII pro plotrovou výrobu DPS, který je specifický tím, že každá vrstva je definována ve vlastním souboru. Pro výrobu je potřeba vyexportovat vrstvy mědi, pájecích plošek, popisky, rozměry desky a soubor k vrtání otvorů. Kromě souborů obsahujících vrtání a rozměry desky jsou potřeba výše zmíněné soubory pro každou stranu desky zvlášť. Pokud by byla deska více vrstvá bylo by potřeba taktéž vyexportovat pro každou vrstvu mědi vlastní soubor.



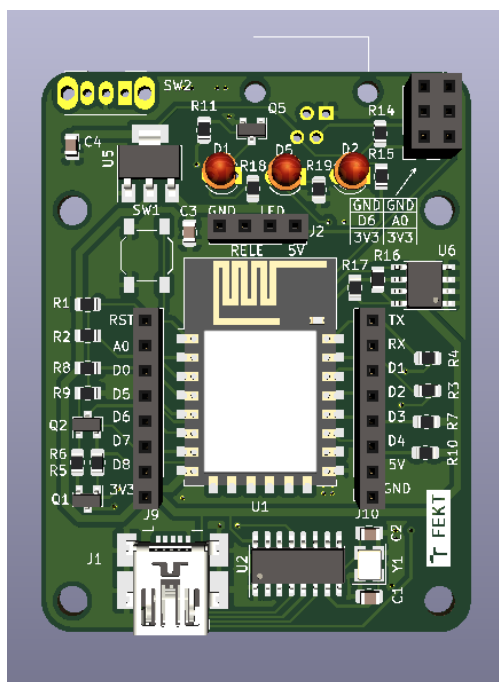
Obrázek 3.6 3D model navržené desky verze 1, export KiCad



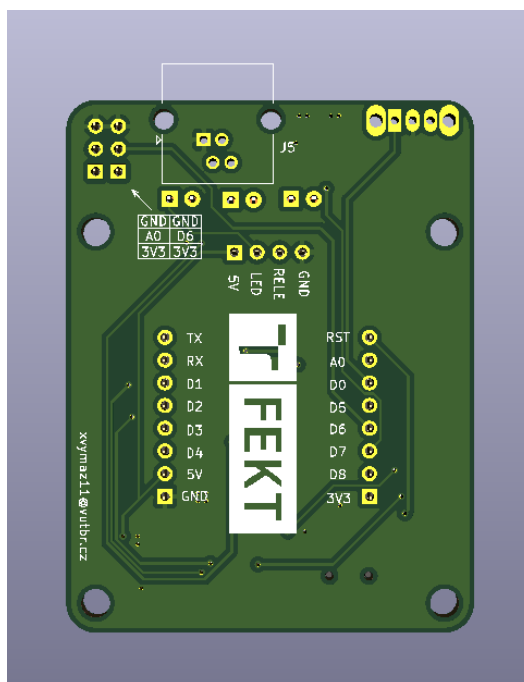
Obrázek 3.7 3D modely DPS verze 2 přední pohled, export KiCad



Obrázek 3.8 3D modely DPS verze 2 zadní pohled, export KiCad



Obrázek 3.9 3D model DPS verze 3 přední pohled, export KiCad



Obrázek 3.10 3D model DPS verze 3 zadní pohled, export KiCad

Zásadním rozdílem mezi verzí 2 a 3 je doplnění chybějící cesty, otočení pinů pro připojení čidel (vpravo nahoře na přední straně). Další úpravou je softwarové ovládání LED diod (zelená, žlutá), dodání popisek a log (změna popisu vývodů Wemos, prohození 5 V a GND) a v posledním případě kompletní změna tras, obzvláště u I2C.

3.6 Osazení a oživení DPS

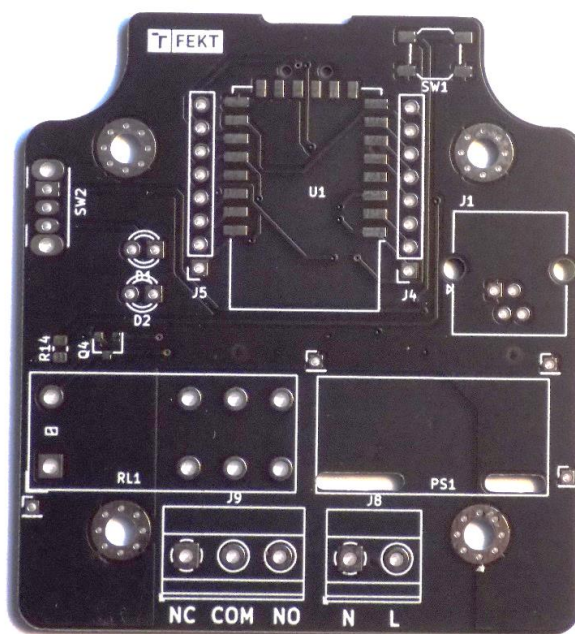
K osazení byl použit bezolovnatý cín. Hlavní motivací k použití bezolovnaté pájky byla směrnice Evropského parlamentu a Rady 2011/65/EU, která zakazuje používání olovnatých pájek v elektrozařízeních, kromě výjimek pro zdravotnictví, kosmický průmysl a vojenské účely [17].

Dalšími nástroji použitými k osazení DPS byla mikropájka a horkovzdušná pistole na osazení krystalu u obvodu USB to UART převodníku.

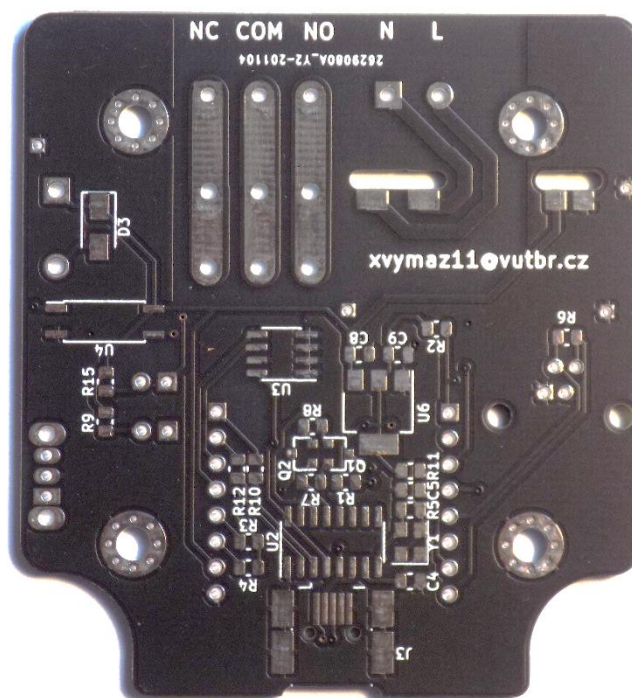
3.6.1 Deska zařízení verze 1

Na obrázku 3.11 a 3.12 je pohled na desku verze 1. Zde jsou připraveny kontakty na připojení relé a svorkovnice, na které bude připojena spínaná zásuvka s proudem 16 A. Z norem pro návrh desek se dá vyčíst, že při tloušťce mědi 35 μm by požadovaná šířka musela mít více jak 2,5 mm. Z toho důvodu se uvažuje přidání pomocného spojení pomocí cínu, a to po celé délce cest. Tohoto propojení pomocí cínové cesty je použito i u ostatních návrhů.

Popis pod konektory pro připojení vodičů nevychází ze značení svorek, ale je zde použito značení podle vodičů. Mnohem vhodnější by bylo místo značení L použít U. Z důvod konzistence bylo na následujících deskách použito taktéž značení vodičů namísto značení svorek.



Obrázek 3.11 Vyrobená deska verze 1 - přední strana



Obrázek 3.12 Vyrobená deska verze 1 - zadní strana

3.6.2 Oživení zařízení verze 1

Osazování desky probíhalo postupně a jednotlivé periferie byly rovnou testovány. Tato metoda pomohla jednoduše nalézt chyby v návrhu. První chyba, na kterou jsem narazil, se projevovala tak, že deska komunikovala s počítačem, ale nešel do mikrokontroleru nahrát firmware. Chybu se mi povedlo odhalit, byla způsobena použitím špatného pouzdra u tranzistorů Q1 a Q2, které mají za úkol při nahrávání programu přepnout ESP do režimu bootu.

Další periferie byly také odzkoušeny. Komunikace s EEPROM pamětí fungovala bez problémů, stejně tak spínání relé. Bohužel komunikace mezi DS18B20 ze začátku nechtěla fungovat. Tato chyba byla způsobena špatnou verzí knihovny.

Osazenou desku verze 1 bylo možné vidět během prezentace semestrální práce. Součástky z této desky bylo potřeba použít pro osazení dalších verzí, z tohoto důvodu není možné předvést osazenou desku verze 1. Z prvního návrhu se zachoval pouze elektronický návrh desky a několik vyrobených desek.

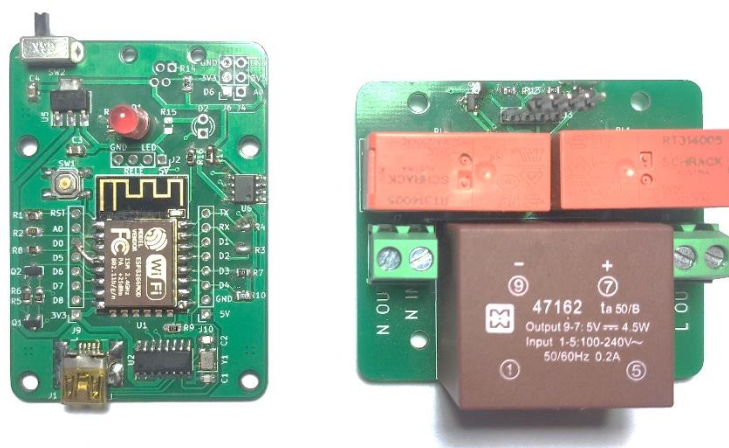
Zajímavostí u této verze desky je její barva, jako jedinou jsem ji nechal vyrobiť v černé barvě, a to kvůli tomu, aby ladila k černému modulu ESP. V dalších verzích jsem od černé barvy upustil a vybral jsem zelenou, aby byla výsledná deska více podobná 3D modelu.

3.6.3 Deska zařízení verze 2

Součástí návrhu zařízení není zdroj. Ve verzi 2 a 3 je použit již zpracovaný zdroj určený k montáži na DPS. Jedná se o zdroj od firmy MYRRA s označením 47162 s výstupním napětím 5 V DC, s maximálním výkonem 4,5 W, s ochranami proti přehřátí, proti přetížení a proti zkratu. Ten byl hlavním důvodem pro návrh nové verze zařízení, které je rozděleno na dvě jednotlivé desky, na zdrojovou a výpočetní desku.

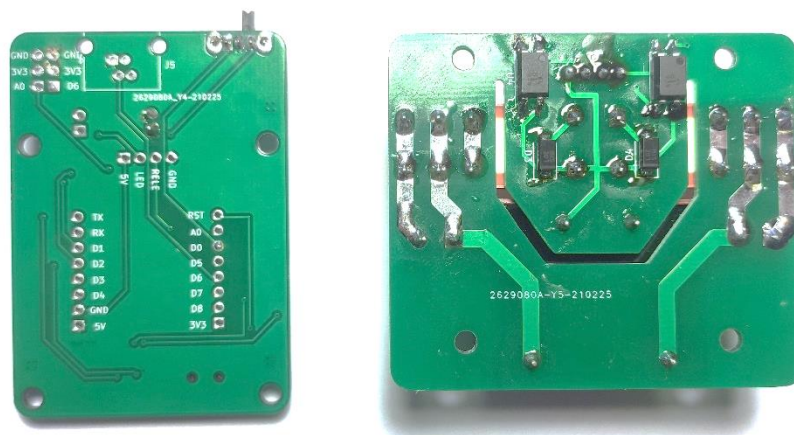
Zdrojová deska obsahuje výše zmíněný MYRRA zdroj, dvě relé a ovládací periferie jako jsou optočleny, tranzistory a pomocné rezistory. Pro spojení s horní výpočetní deskou je použit hřebíkový konektor se čtyřmi piny (GND, RELE, LED, 5 V). GND a 5 V je napájení pro řídicí – výpočetní desku. Signálem RELE je možné ovládat spínání relátek na zdrojové desce (3,3 V zapnuto a 0 V vypnuto). Signál LED je uzemněn, pokud jsou relé sepnuty. Díky tomu může spínat červenou LED diodu na výpočetní desce, která indikuje sepnutí výstupní zásuvky.

Výpočetní deska obsahuje zbylé komponenty: modul ESP, EEPROM, stabilizátor, USB převodník a kontrolní LED diody. Desky jsou spojeny pomocí konektorů a speciálního distančního sloupku, viz obrázek 3.15.



Obrázek 3.13 Vyrobená a osazená deska verze 2 – přední strana

Na obrázku 3.13 je pohled na přední stranu desek verze 2, vlevo se nachází výpočetní deska, na které je osazena pouze jedna signalizační LED dioda a chybějící cesta je improvizovaně nahrazena drátkem (levá strana modulu ESP). Vpravo se nachází zdrojová deska, která je stejná i pro verzi 3. Na zdrojové desce lze vidět MYRRA zdroj (velká hnědá krabička), dvě relátka (dva oranžové objekty) a dva konektory pro připojení síťového napájení.



Obrázek 3.14 Vyrobená a osazená deska verze 2 – zadní strana

3.6.4 Oživení zařízení verze 2

Při ožiování zdrojové desky nebyly nalezeny žádné problémy, vše bezproblémově fungovalo. Oproti tomu na výpočetní desce chyběla cesta od přepínače do MCU, kterou jsem nahradil propojkou. Další problémem byly LED diody, které sice svítily, ale vůbec nic neříkaly o tom, v jakém stavu se nachází program. Při špatném wifi signálu se může stát, že ESP není schopné se připojit do sítě, z toho důvodu je potřeba indikovat nejen do kterého režimu je výrobek přepnut, ale jestli se ESP připojilo. Při poruše čidla nebo při rozbití poškození kabelu čidla je potřeba, aby ESP hned signalizovalo poruchový stav. Proto byla další verze o tyto indikační LED diody rozšířena.

V neposlední řadě jsem se zaměřil na externí EEPROM paměť (M24512), kterou jsem následně nahradil 24LC16B. Důvod, proč korektně nefungovala EEPROM paměť M24512 přikládám tomu, že se pravděpodobně jednalo o vadnou dávku. Ze tří pamětí, které jsem měl, se každá chovala jinak. Jedna nefungovala hned od začátku, další fungovala chvíli a pak přestala zapisovat a poslední nechtěla zapisovat na určitá místa. Prvotním plánem byla myšlenka, že by veškeré textové proměnné (texty na webu) byly taktéž uloženy v EEPROM paměti a za běhu programu by se dal editovat obsah webu. Tato myšlenka se později ukázala jako zbytečná.

Další, spíše kosmetickou chybu jsem našel na popisku Wemos rozhraní, kde byly prohozeny popisky na 5 V a GND. V následující verzi jsou popisky už správně. Ne zcela zjevnou změnu prodělali i piny pro připojení digitálního a analogového čidla (vpravo nahoře při čelním pohledu). Prohozením pořadí těchto výstupů jsem umožnil elegantnější připojení DS18B20 v pouzdře TO92.

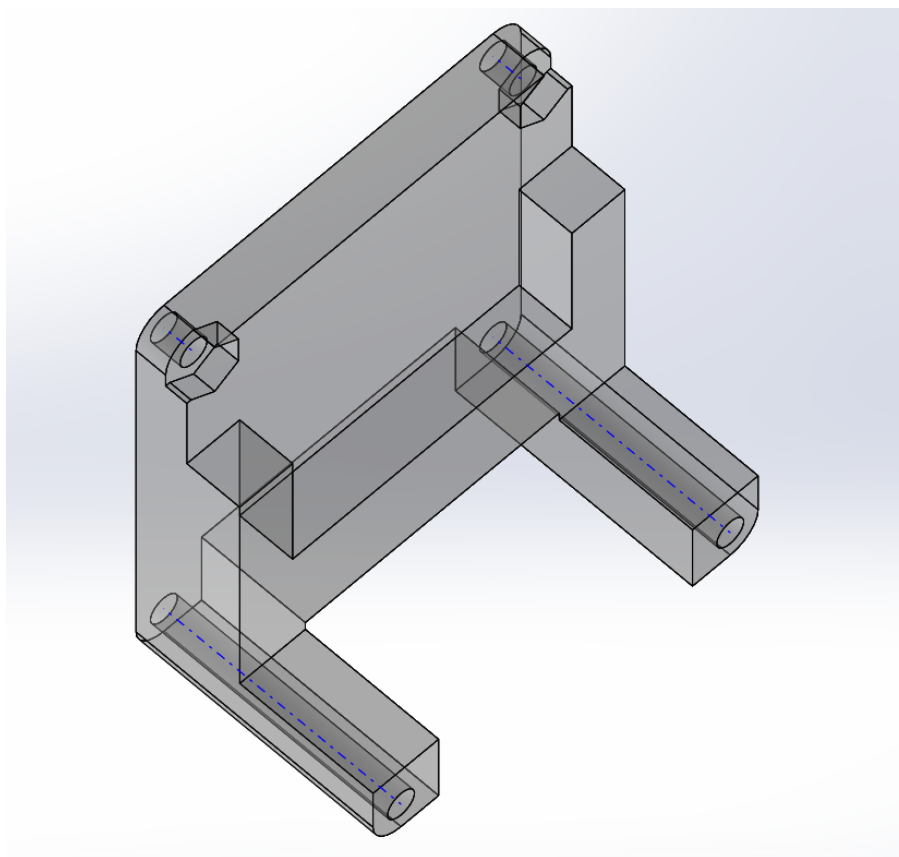
Nejzávažnější problémem, proč nemohla být použita výpočetní deska verze 2, byla její velikost. Při pokusu dostat celé zařízení do krabičky jsem zjistil, že deska přesahuje

asi o 3,5 mm mimo prostor krabičky. Nejjednodušším a nejelegantnějším řešením bylo vytvořit novou verzi.

Při té příležitosti jsem vytvořil speciální distanční sloupek, pomocí kterého jsem mohl desky k sobě uchytit a tím zjistit, o kolik musím novou desku zkrátit. Tento sloupek se mi tak osvědčil, že jsem se rozhodl ho zakomponovat do celého výroku, aby byly desky pevně přichyceny k sobě.

K návrhu distančního sloupku bylo potřeba nejprve osadit celou zdrojovou desku, následně provést měření a z naměřených data vytvořit v programu SOLIDWORKS 3D model. Tento model je potřeba uložit ve formátu STL, aby mohl být dále zpracován. Pro převod STL modelu na G-code byl použit program SLIC3R, nejzásadnější konfigurací pro tento slicer bylo nastavení výšky vrstvy na 0,2 mm a výplně na 30 %. Sloupek byl vytištěn z materiálu PLA při teplotě 205 °C.

Protože při použití tohoto distančního sloupku by mohlo dojít k akumulaci tepla především na relátkách, nechal jsem jednu stranu co nejvíce odkrytou, aby zde mohlo docházet k úniku tepla do okolí. Hned vedle relátek je umístěn zdroj, který se od nich bude taktéž částečně zahřívat. Tento zdroj je vybaven tepelnou ochranou, takže by se při zahřátí na 50 °C měl odstavit, čímž by došlo i k vypnutí relátek, a zařízení by tak mělo za čas zchladnout, dokud by zdroj zase nezačal fungovat.

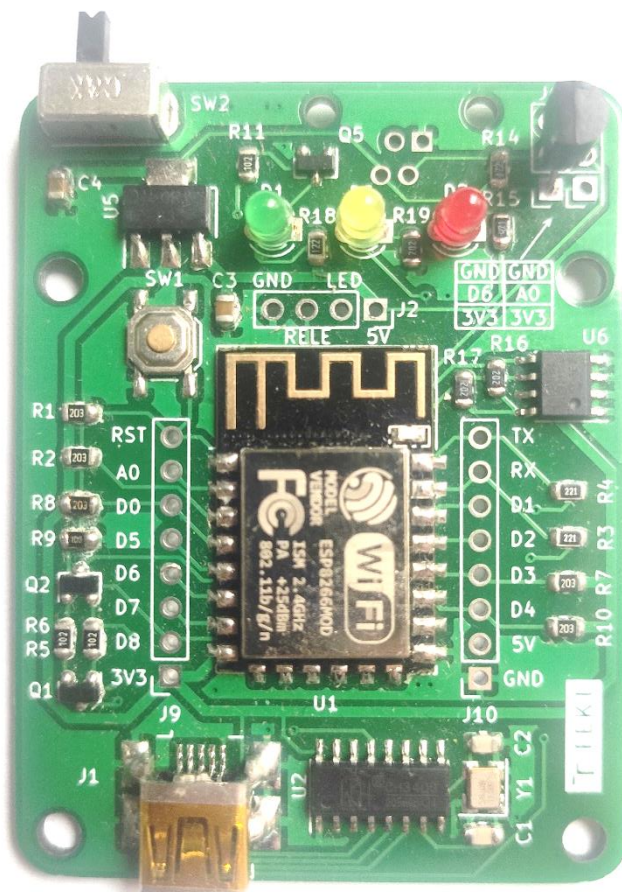


Obrázek 3.15 Speciální distanční sloupek pro propojení desek

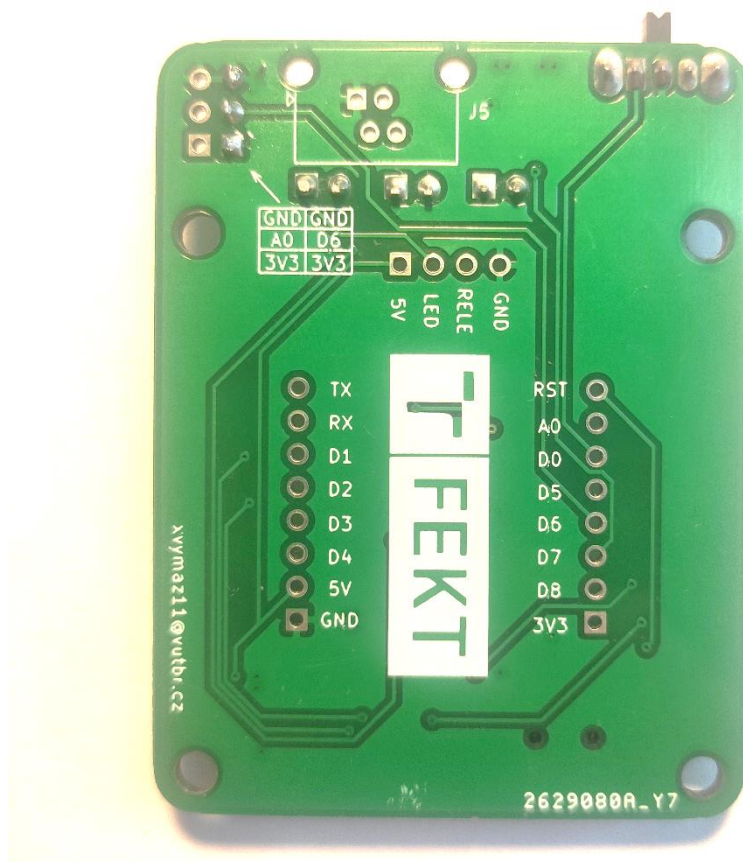
3.6.5 Deska zařízení verze 3

V poslední 3. verzi se povedlo odstranit veškeré problémy, na které jsem narazil při testování předchozích verzí. Při rychlém pohledu se může zdát, že se tato poslední verze příliš neliší od svého předchůdce, ale opak je pravdou. Jediné, co bylo zachováno je přibližné rozmístění součástek. Občas bylo potřeba součástku pootočit nebo nepatrně posunout. Na první pohled je ve třetí verzi rozdílný počet LED diod, ta zásadnější změna je v tom, že zelenou a žlutou diodu je možné ovládat softwarově.

Na obrázku 3.16 lze vidět již osazenou desku verze 3, kde místo RJ konektoru pro připojení čidla je DS18B20 přímo připájen na vývody vpravo nahoře. Důvodem je, aby deska byla skladnější při následném vývoji firmwaru. V budoucnu je také možnost zde připojit další interní čidlo, které by mohlo sledovat teplotu uvnitř krabičky. Hned vedle jsou další pájecí body, kde je možnost připojit analogové teplotní čidlo.



Obrázek 3.16 Vyrobená a osazená výpočetní deska verze 3 – přední strana



Obrázek 3.17 Vyrobená a osazená výpočetní deska verze 3 – zadní strana

3.6.6 Oživení zařízení verze 3

Při oživování finální verze desky se objevil pouze jeden problém, který se povedlo poměrně rychle vyřešit. Při snaze co nejméně snížit spotřebu zařízení jsem se rozhodl použít místo 20 k Ω rezistorů 200 k Ω rezistory, bohužel tato změna způsobila problém: ESP nechtěl nastartovat a donekonečna se restartoval. Při vrácení 20 k Ω rezistorů tato závada zmizela.

U této verze desky také bylo potřeba správně navrhnout rezistory u LED diod. Protože jsem použil diody, které jsem měl doma, bylo potřeba stanovit jejich pracovní bod. K tomu byl využit regulovatelný zdroj a ampérmetr. K diodě v sériovém zapojení s ampérmetrem byl připojen regulovatelný zdroj. Postupným zvyšováním napětí na zdroji, dokud se dioda nerozsvítila s požadovanou intenzitou, jsem zjistil potřebné informace o pracovním bodě. Především se jednalo o napětí na diodě U_d a o velikost proudu I_d , který protékal diodou. Tato metoda byla opakována pro všechny barvy použitých LED diod. Naměřená data jsou uvedena v následující tabulce 3.2.

Tabulka 3.2 Tabulka změřených pracovních bodů LED diod

Barva LED diody	Úbytek na diodě [V]	Proud diodou [mA]	Referenční napětí [V]
červená	1,7	7	5
žlutá	1,8	0,7	3,3
zelená	3,2	1,5	5

Výpočet potřebné hodnoty rezistoru jsem provedl podle následujícího vzorce 3.4, ve kterém referenční napětí U_r je napěťová úroveň, ze které bude LED dioda napájena. R značí hodnotu hledaného rezistoru, I_d a U_d značí proud a napětí potřebné k tomu, aby se dioda rozsvítila.

$$R_{barva} = \frac{U_r - U_d}{I_d} \quad (3.4)$$

Postupně jsem provedl výpočty pro všechny použité barvy. Protože rezistory jsou vyráběny pouze v určitých hodnotových řadách je nutné získaný výsledek převést na hodnotu rezistoru, která je běžně vyráběna. Díky tomu sice dojde k posunu pracovního bodu, ale tahle změna proudu a napětí bude zanedbatelná.

$$R_{červená} = \frac{U_r - U_d}{I_d} = \frac{5 - 1,7}{0,007} = 471,43 \, \Omega \quad (3.5)$$

$$R_{žlutá} = \frac{U_r - U_d}{I_d} = \frac{3,3 - 1,8}{0,0007} = 2142,86 \, \Omega \quad (3.6)$$

$$R_{zelená} = \frac{U_r - U_d}{I_d} = \frac{5 - 3,2}{0,0015} = 1200,00 \, \Omega \quad (3.7)$$

Z vypočtených hodnot jsem vybral pro červenou diodu hodnotu $470 \, \Omega$, pro žlutou $2000 \, \Omega$ a pro zelenou $1200 \, \Omega$. Po osazení a otestování vše fungovalo bez problému.

U všech verzí desky je pro připojení teplotního sensoru použit konektor RJ13. Následující tabulka popisuje zapojení čidla k RJ konektoru dle standartního číslování.

Tabulka 3.3 Tabulka zapojení teplotního čidla k RJ konektoru

Číslo pinu JR konektoru	Vývod čidla
1	GND
2	DATA
3	nepřipojeno
4	3,3 V – napájení VCC

4. FIRMWARE

Pro projekt byl použit programovací jazyk C/C++. Pro usnadnění práce bylo použito Arduino studio, obzvláště při kompilaci a uploadu firmwaru. Kód byl napsán v textovém editoru Atom, Arduino studio není nejvhodnější textový editor pro vývoj větších projektů, přesto velice usnadňuje kompilaci a nahrání.

Sofwarová část projektu byla rozdělena na dílčí části, aby byla přehlednější a snadno editovatelná. Snahou také bylo využít výhod objektového programování. Projekt je členěn do funkčních bloků (tříd), každá třída má za úkol jinou specifickou činnost.

Požadavky na tento projekt byly regulace teploty, ukládání naměřených dat na IoT cloud, možnost přepínat v režimech wifi AP a wifi klient a možnost ovládat a konfigurovat zařízení pomocí webového rozhraní.

Základní strukturu firmwaru lze vidět na následujícím obrázku 4.1., na kterém jsou vidět dva základní bloky, ze kterých se skládá většina firmwaru. Jedná se o část kódu, která se vykoná pouze jednou. V main souboru ji lze najít pod označením setup. Druhá část kódu pojmenovaná loop je vykonávána stále dokola dokud nedojde k vypnutí nebo resetování zařízení.

Při prvním nahrávání programu do ESP je potřeba ve třídě *Config* v metodě begin zakomentovat načítání dat z EEPROM a definovat data potřebná pro spuštění napevno v kódu. Následně je potřeba načíst webovou stránku a tam provést uložení dat na EEPROM. Po uložení je nutné do zařízení nahrát program, kde jsou smazány napevno nastavené hodnoty sítě a je povoleno načtení těchto dat z EEPROM. Důvodem postupu je, že nová EEPROM neobsahuje konfiguraci webového rozhraní a došlo by k načtení nesmyslných dat, se kterými by zařízení nepracovalo korektně a spolehlivě. Výpis bloku kódu z metody begin ze třídy Config je ukázán níže.

```
/****** KOD POTŘEBNÝ PRO PRVNÍ NAHRÁNÍ FIRMWARE *****/
setNAME("esp8266");
setSSID("ESP8266");
setPASS("rootroot");
setIP("192.168.10.1");
setMASK("255.255.255.0");
setGATEWAY("192.168.10.1");

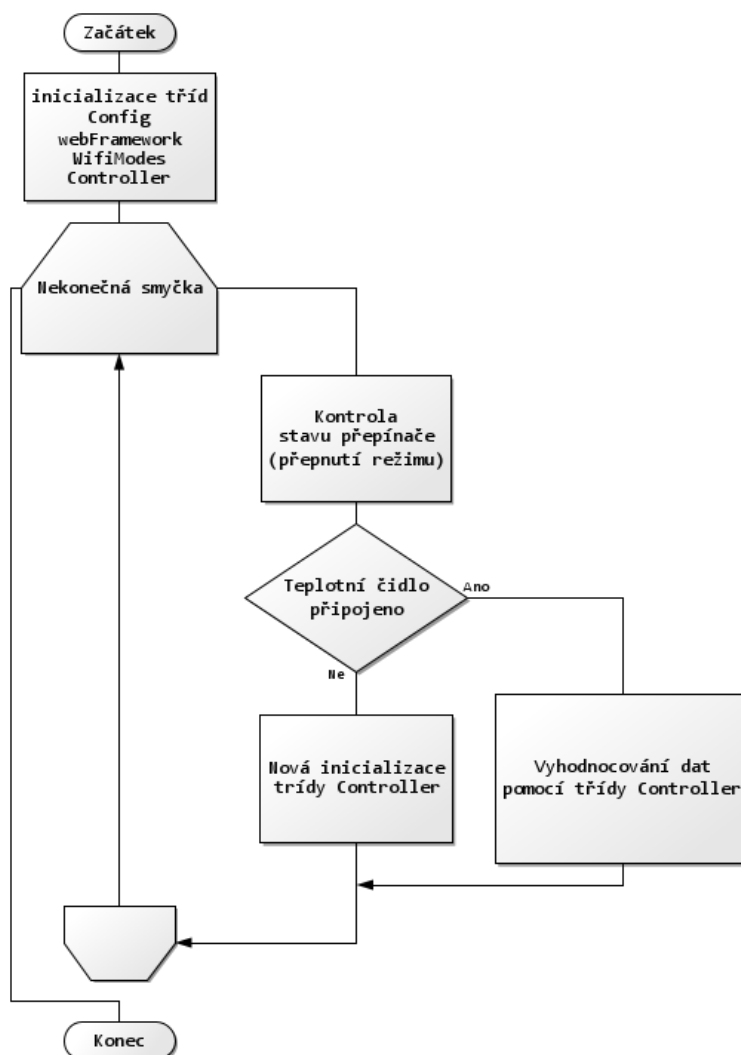
STATION_addConnection("SSID", "password");
STATION_addConnection("SSID2", "password2");

setTimerLogger(600000);
setMyChanelNumber(1365234);
setApiKey("RR68JA802JYI63K8");

//load();          // Odkomentovat při druhém nahrávání a smazat zbytek
```

Princi měření teploty je ukázán na následujícím vývojovém digramu Obrázek 4.1, popisující strukturu hlavního main souboru. Následné měření a vyhodnocování dat je prováděno pomocí třídy Controller.

Rozhodnutí, jestli má být výstupní zásuvka sepnuta je implementován ve třídě Controller v metodě _control. K sepnutí zásuvky dojde, pokud je měřená teplota nižší než nastavená teplota navýšená o polovinu hodnoty hystereze. Jakmile čidlo dosáhne požadované teploty navýšené o polovinu hystereze, dojde k odpojení zásuvky. Zásuvka je znovu sepnuta, pokud teplota klesne pod nastavenou hodnotu o polovinu hodnoty hystereze. Hystereze je v tomto způsobu regulace především použita k omezení počtu sepnutí a rozepnutí relé, neboť každé relé má definovanou svoji životnost počtem sepnutí. Díky tomu je docíleno vyšší životnosti zařízení.



Obrázek 4.1 Diagram popisující základní strukturu firmwaru [21]

4.1 Použité knihovny

V této kapitole se zabývám knihovnami, které byly použity při vývoji firmwaru doplněných o krátký popis. V převážné většině se jedná o běžně používané knihovny, se kterými se můžeme setkat mnoha dalších projektech. Jedinou výjimkou je knihovna pro vytvoření asynchronního webového serveru.

Arduino.h

Jedná se o základní knihovnu pro Arduino core, obsahuje základní definice jednoduchých funkcí, jako je digitalWrite nebo digitalWrite. Součástí jsou také definice jmen pinů. Tato knihovna je sice automaticky přidávána, ale pouze do main souboru. Pokud je potřeba napsat knihovnu pracující s jejími funkcemi, musíme ji přidat.

ESP8266WiFi.h

Tato knihovna umožňuje přístup k wifi periférii. Především se přes ni nastavuje síťové rozhraní a režim, v jakém má zařízení fungovat. Díky této knihovně je možné i získávat data o okolních wifi sítích.

ESP8266WiFiMulti.h

Jde o nadstavbu nad ESP8266WiFi.h, díky této knihovně je možné nastavit seznam wifi sítí, na který se má připojit. Knihovna se připojí k první možné wifi síti ze seznamu.

WiFiClient.h

Pomocí této knihovny je možné realizovat TCP/IP spojení. V projektu je především použita jako podpora pro ThingSpeak.h knihovnu, díky které je možné ukládat data na IoT cloud.

ESP8266mDNS.h

Tato knihovna realizuje multicast DNS server, díky němuž je možné zařízení nalézt v místní síti pomocí jeho mDNS jména. V reálných sítích a na některých zařízeních tenhle protokol nefunguje. Přesto byla snaha ho implementovat.

Wire.h

Knihovna implementující I2C/TWI komunikaci, která je potřeba pro komunikaci s EEPROM pamětí.

OneWire.h

Implementuje 1-Wire sběrnici, jedná se o upravenou I2C sběrnici. Díky této knihovně je možné komunikovat s DS18B20 čidlem.

stdint.h

Jde o standardní C knihovnu, v projektu je především použita pro import datových typů jako je například uint8_t atd.

stdarg.h

Jedná se o standardní C knihovnu, díky níž je možné vytvořit funkce s proměnným počtem parametrů.

DallasTemperature.h

Knihovna realizující interface mezi uživatelem a OneWire knihovnou při komunikaci s teplotním čidlem DS18B20. Knihovna především obsahuje funkce pro vyčítání hodnot a převod mezi °C a °F.

ThingSpeak.h

Jedná se o rozhraní, které ulehčuje komunikaci se stejnojmenným IoT cloudem.

ESPAsyncTCP.h

Málo rozšířená knihovna pro asynchronní TCP dotazy, je využívána především níže popsaným webovým serverem, který ji využívá pro svoji práci.

ESPAsyncWebServer.h

Rozsáhlá knihovna realizující webový server, jedná se o málo rozšířenou knihovnu realizující webový server na ESP2866, ale knihovna má velice podrobně zpracovanou technickou dokumentaci se spoustu ukázek. Velkou výhodou toho serveru je jeho funkcionality a jednoduchost.

FS.h

Umožňuje zprávu a přístup do virtuálního souborového systému, který je možné na ESP vytvořit a provozovat. V tomhle souborovém systému jsou například uloženy HTML šablony, CSS soubory a veškeré JavaScript funkce, které jsou využívány webovým rozhraním.

LittleFS.h

Rozšiřuje funkcionality FS knihovny.

4.2 Vytvořené knihovny

webFramework.h

Knihovna starající se o načítání a webové stránky, zasílání JSON dat a odchytávání GET a POST dotazů.

ProjectConfig.h

Soubor, ve kterém jsou uloženy globální proměnné potřebné pro sestavení projektu, například definice jmen pinů a deklarace funkcí usnadňují tisk do konzole.

Config.h

Jedná se o knihovnu, ve které jsou uloženy veškeré proměnné potřebné pro běh zařízení, síťová konfigurace, mDNS jméno atd.

WifiModes.h

Stará se o nastavení wifi rozhraní, připojuje se do sítě nebo vytváří vlastní wifi síť, a to v závislosti na tom, do jaké pozice je přepnut přepínač na DPS. V této knihovně je také implementován mDNS server.

LList.h

Speciální knihovna realizující spojovou strukturu, pro ukládání wifi sítí, kterým se má zařízení připojovat.

EEPROM_24LC16B.h

Implementuje rozhraní pro práci s EEPROM pamětí.

Controller.h

Jde o knihovnu zodpovědnou za vyčítání teploty a rozhodnutí, jestli má být vstupní zásuvka sepnuta. Dále zodpovídá za odesílání dat do IoT cloudu.

Buffer.h

Pomocná knihovna umožňující tisk do bufferu, především pro zjednodušení výpisu dat na konzoli.

Timer.h

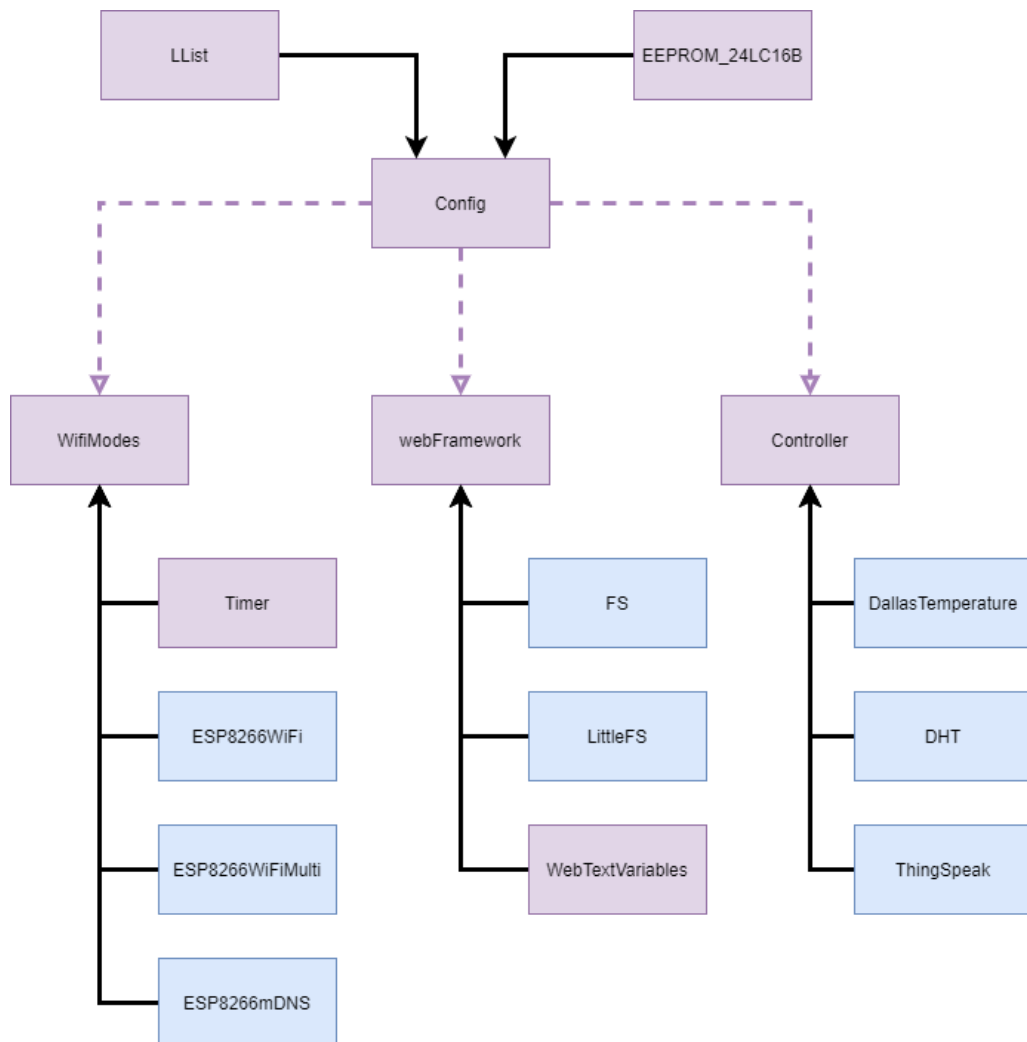
Umožňuje časovat akce, které je potřeba spustit jednou za dlouhý časový úsek.

WebTextVariables.h

V toto souboru je uloženo pole, ve kterém jsou definovány veškeré textové proměnné, které jsou vypisovány na webové stránce.

4.3 Závislosti mezi třídami

Pro projekt je sice využito, objektového jazyka C++, ale třídy jsou zde použity pouze jako funkční bloky. Z tohoto důvodu vznikly závislosti při vytváření jednotlivých instancí. Tyto závislosti jsou vizuálně zobrazeny na Obrázku 4.2, kde je potřeba rozlišit, jestli dochází k předávání odkazu při vytváření nové instance nebo k vytvoření instance v instanci. Případ kdy třída vyžaduje odkaz na již vytvořenou instanci je na obrázku naznačen přerušovanou fialovou čarou. Druhý případ, kdy třída vytváří instanci jiné, je definován černou plnou čarou. Třídy jsou dále rozděleny barevně podle toho, kým byly vytvořeny. Modrá barva značí, že se jedná o knihovny, které jsou součástí základních knihoven pro Arduino nebo volně šiřitelných knihoven třetích stran. Fialové bloky značí knihovny a moduly, které bylo potřeba vytvořit speciálně pro tento projekt.



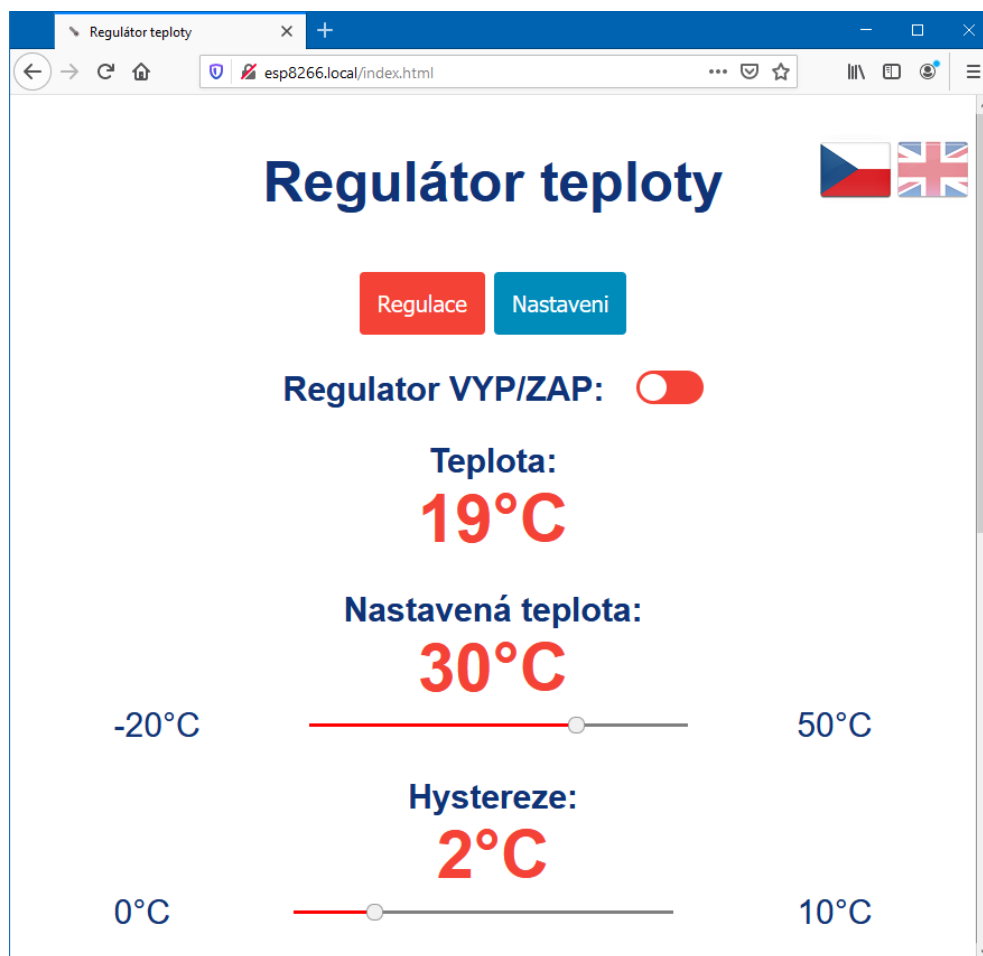
Obrázek 4.2 Struktura popisující závislosti mezi použitými třídami [22]

5. ROZHRAŇÍ ZAŘÍZENÍ

5.1 Webové rozhraní

Pro konfiguraci a vizualizaci naměřených data je možné použít webové rozhraní. Jedná se o dvě jednoduché webové stránky, na první z nich jsou zobrazena měřená data a také je zde možné nastavit parametry potřebné pro regulaci, jako jsou například požadovaná teplota, hystereze regulace a možnost kompletně odstavit regulaci. Na této stejné stránce je v režimu klient zobrazován graf s naměřenými hodnotami. Další zajímavostí je grafická vizualizace signalizačních LED diod, které nám mohou usnadnit určení, v jakém stavu se zařízení právě nachází.

Na další webové stránce je možné provádět konfiguraci síťového rozhraní, konfiguraci spojení na IoT cloud nebo zprávu wifi sítí, ke kterým se může ESP připojovat. Pro uložení takto nastavených dat je potřeba změnu konfigurace uložit na EEPROM. Toho se docílí tlačítkem, které se taktéž nachází na této stránce, viz následující obrázek 5.1.



Obrázek 5.1 Úvodní stránka webového rozhraní

5.2 Sériová linka

Pro vizualizaci dat a logování je v této bakalářské práci použit výpis do terminálu pomocí sériové linky. V operačním systému MS Windows můžeme k sériové lince přistoupit pomocí COM portů, v Linuxových operačních systémech pomocí virtuálního souboru ttyS. Za každým z těchto označení (COM nebo ttyS) je uvedeno ještě číslo definující jeden port, například COM4 nebo ttyS2. Aby terminál byl schopen správně přijímat data, je potřeba při připojování definovat rychlost komunikace. V anglické literatuře se označuje jako *baud rate*, který nám říká kolik znaků jsme schopni přijmout nebo odeslat za jednu sekundu.

V případě tohoto zařízení není možné zařízení ovládat nebo konfigurovat pomocí terminálu, je možné pouze přijímat/logovat akce, které se v zařízení odehrávají. Pro projekt je možné použít terminál, který je součástí Arduino IDE nebo například program PuTTY.

```
COM5
Chip ID.....10709140
Chip info Flash Chip ID.....1458392
Chip info Core version.....2_7_4
Chip info SDK version.....2.2.2-dev(38a443e)
Chip info Boot version.....6
Chip info CPU frequency.....80 MHz
Chip info FREE HEAP.....41960 B
Chip info FREE CONST. STACK.....3120 B
Chip info real FLASH size.....4194304 B
-----
WifiConfig select Mode.....STATION (Client)
WifiConfig Multi Wifi list connecting.....RUN
WifiConfig STATION MODE SSID.....UPC_2G
WifiConfig STATION MODE IP.....192.168.0.227
-----
mDNS responder started.....OK
DallasTemperature find first device.....OK
DallasTemperature read address.....OK
Controller RELE.....ON
Controller RELE.....OFF
Temperature sensor.....Disconnected
DallasTemperature find first device.....OK
DallasTemperature read address.....OK
Controller RELE.....ON
Controller RELE.....OFF
IoT Cloud send.....OK
IoT Cloud send.....OK
```

Obrázek 5.2 Výpis akcí provedených zařízením vypsaných v terminálu

Z obrázku 5.2 je vidět co zařízení provedlo od svého startu. Na prvním řádku je vidět text, který je typický pro výpis z ESP. Podle tohoto výpisu poznáme start modulu a inicializaci sériové linky. Následuje blok dat, které nám říkají základní informace o modulu a jeho paměti. Za ní je umístěn blok informující o tom, do jakého síťového režimu je zařízení přepnuto, jakou má ip adresu a do jaké wifi sítě je připojeno. V posledním bloku dat je záznam o detekci čidla následovaném sepnutím výstupní zásuvky. Po dosažení požadované teploty byla spínaná zásuvka odpojena. Na dalším řádku je záznam informující o odpojení čidla. Poslední řádek informuje o zaslání dat na IoT cloud.

5.3 Signalizační diody

Posledním a zároveň nejdůležitějším rozhraním jsou signalizační LED diody, které jsou umístěny na DPS a pomocí otvorů v krabici vyvedeny až ven. Zařízení obsahuje tři diody s následujícími barvami: zelená, žlutá a červená. Podle chování těchto diod můžeme detekovat poruchové stavy nebo jednoduše určit v jakém režimu se zařízení nachází nebo jestli se připojilo do místní sítě.

Červená LED dioda značí, jestli jsou sepnuta výstupní relé neboli jestli je na výstupní zásuvku přivedeno 230 V. Volba červené diody nebyla náhodná, volba barvy byla inspirována barevným označením jističe v sepnutém stavu. Na rozdíl od jističe došlo k malé změně při indikování vypnutého stavu. V tohle případě je signalizován tak, že červená LED nesvítí.

Žlutá LED dioda oznamuje poruchový stav nebo start zařízení. Zelená dioda indikuje stav sítě a režim zařízení, ve kterém se nachází. Při startu zařízení v režimu klient se nejprve rozsvítí žlutá dioda, zelená dioda je vypnutá, pouze při nezdařilém pokusu o připojení se zelená dioda překlápí do opačného stavu. V tom případě bude zelená dioda blikat velice nepravidelně. Režim AP je indikován blikáním zelené diody (frekvence 0,5 Hz) bez toho, aby se rozsvítila i žlutá dioda. Jednotlivé stavy LED s popisy stavů, jsou uvedeny v následující tabulce 5.1, kde pomlčka značí, že na daném stavu diody nezáleží.

Tabulka 5.1 Signalizace LED diodami

Červená	Žlutá	Zelená	Stav
svítí	-	-	Zásuvka sepnuta
nesvítí	-	-	Zásuvka odpojena
-	svítí	nesvítí/bliká	Připojování do wifi sítě
-	-	bliká	Režim AP
-	-	svítí	Režim Klient
-	svítí	-	Poruchový stav

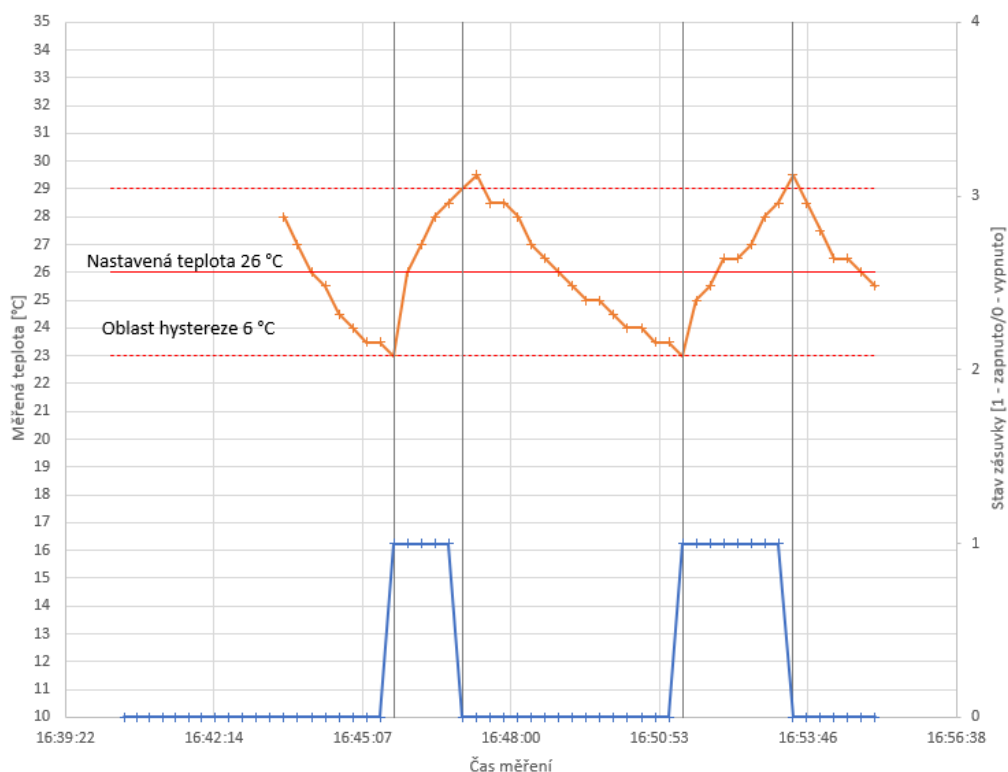
Při zvoleném způsobu indikace může vzniknout stav, který není úplně jednoznačný. Jedná se o situaci, kdy zelená dioda bliká a žlutá svítí, buď se jedná o stav, kdy je zařízení v režimu AP a je odpojeno čidlo nebo se jedná o nezdaření připojení do sítě v režimu klient. Každopádně se vždy jedná o chybový stav, pouze příčina je jiná. Řešením této situace je reset zařízení a pokud bude po startu zelená dioda déle jak 2 sekundy beze změny jedná se o situaci, kdy se zařízení snaží připojit do sítě.

Další záludností je červená dioda. Regulace startuje hned po spuštění zařízení, zato připojení do sítě může trvat déle, proto je naprosto normální, že se jako první rozsvítí červená dioda a dojde k sepnutí zásuvky. Aby k této situaci došlo, tak musí být v regulátoru povolena regulace a připojeno teplotní čidlo. Každopádně se nejedná o chybový stav nebo nějakou chybu.

6. OVĚŘENÍ FUNKČNOSTI REGULACE

Na následujícím Obrázku 6.1 lze vidět graf, na kterém je ukázáno, jak probíhá regulace pomocí navrženého zařízení. Dolní modrý průběh ukazuje, v jakém stavu se nacházela spínaná zásuvka. Hodnota jedna odpovídá sepnuté zásuvce, hodnota nula znamená vypnutou zásuvku. Oranžově je zobrazena teplota měřená teplotním čidlem a červenou plnou čarou je naznačena požadovaná nastavená teplota, v tohle případě na 26 °C. Oblast hystereze kolem červené plné čára je naznačena červenými přerušovanými čarami. V měřeném případě byla hystereze nastavena na hodnotu 6 °C, hlavně z toho důvodu, aby graf měl patřičnou vypovídající hodnotu.

Na grafu jsou vidět dva cykly sepnutí zásuvky. Graf průběhu teploty je závislý na několika faktorech. Uvažujeme-li prostředí, ve kterém zanedbáme ohřev slunečním zářením, bude takové prostředí pomalinku chladnout. V grafu se projeví jako oranžová klesající křivka. Jakmile teplota klesne na krajní hodnotu hystereze, topení se sepne, nastane pomalý nárůst teploty. V grafu je vidět jako rostoucí úsek oranžové křivky. Až teplota vystoupá na horní okraj hystereze, zásuvka se vypne a odpojí topení a nastává opět fáze chladnutí. Tímto způsobem topení nikdy nemůže docílit požadované teploty, ale dokáže udržovat teplotu, která je blízká požadované teplotě, což pro účel termostatu bohatě dostačuje.



Obrázek 6.1 Graf nespojité regulace vytvořený v programu Excel

7. ZÁVĚR

Bakalářská práce byla rozdělena do dvou částí. V teoretická části se zabývám způsoby měření teploty. Teorie se především zaměřuje na teplotní čidla a IoT cloudy z pohledu použitelnosti pro vyvíjené zařízení. Ze zmíněných teplotních čidel byly vybrány DS18B20 a DHT22. Z IoT cloudů byl jako nejvhodnější vybrán ThingSpeak a to hlavně kvůli dobré dokumentaci a jednoduchosti jeho nasazení.

V praktické části se práce věnuje vývoji zařízení, které bylo doprovázeno třemi verzemi DPS. Poslední finální verze je vybavena firmwarem, jehož hlavní struktura a myšlenky byly nastíněny v této práci. Díky tomuto firmwaru zařízení dokáže za běhu detekovat libovolné ze dvou zmíněných čidel, stejně tak jejich odpojení. Dále implementuje webové rozhraní, pomocí kterého lze regulátor ovládat a konfigurovat. Zařízení nabízí provoz ve dvou režimech podle toho, jestli je možnost zařízení připojit k wifi síti. Pokud je zařízení připojeno do sítě, která má přístup do veřejného internetu, je rozšířena funkcionality o odesílání dat na výše zmíněný IoT cloud.

Firmware je navržen tak, aby zařízení bylo odolné proti ztrátě dat důsledkem výpadku distribuční sítě. Firmware byl napsán s velkým důrazem na uživatelský komfort z toho důvodu jede ESP8266 na hranici svých možností. Z tohoto důvodu by byla vhodná výměna za ESP32, které je výkonnější a regulátoru by poskytlo dostatečný výpočetní výkon.

Cílem regulátoru je ovládat okolní teplotu pomocí elektrického vytápění. Finální výrobek je spíše určen do prostředí, které pomalu chladne a dlouho se vytápí. Reakční doba na změnu teploty je kolem jedné sekundy, proto není vhodný do aplikací, kde se rychle mění teplota. Zato jako regulátor pro vytápění jedné menší místnosti nebo pro spínání centrálního vytápění je naprosto ideální.

Projekt by v budoucnu mohl být rozšířen o externí anténu, která by mohla vyřešit problém s malým dosahem integrované antény. Dalším krokem by mohlo být vytvoření aplikace na mobilní telefon, která by mohla nahradit webové rozhraní. Díky tomu by se snížily požadavky na výpočetní výkon ESP čipu a značně by to usnadnilo uživatelský přístup. Častým problémem je nefunkčnost mDNS. Při vývoji bylo zjištěno, že tento protokol nefunguje na android zařízeních.

Bezpečnost zařízení po softwarové stránce také není součástí této práce. Z toho důvodu by bylo v budoucnu vhodné rozšířit webovou stránku o SSL/TLS certifikát. Dále by bylo vhodné webovou stránku rozšířit o autentizaci, aby byl omezen počet lidí, kteří smí ovládat termostat.

Výstupem této práce je zařízení, jehož fotka je v příloze tohoto dokumentu. U vyvinutého zařízení se povedlo docílit všech požadavků ze zadání. Díky tomu se jedná o relativně bezpečný a funkční přístroj, který vhodný pro regulaci elektrického odporového vytápění do výkonu 3,6 kW.

LITERATURA

- [1] BRZOBOHATÝ Lukáš: *Měření teploty*. Brno, 2011. 19s., CD. FSI VUT v Brně, Ústav strojírenské technologie, Vedoucí práce Ing. Kamil Podaný, Ph.D.
- [2] *Čidla teploty Pt 100: datasheet* [online]. SENSIT HOLDING [cit. 2020-12-07]. Dostupné z: <http://remaxcz.com/old/katalog/snimace/teplota/kl/char/Pt100.pdf>
- [3] EARNSHAW, Alan a Norman Neill GREENWOOD. *Chemie prvků*. Svazek I. Přeložil František JURŠÍK. Praha: Informatorium, 1993. ISBN 80-85427-38-9.
- [4] *Polovodiče Termistory: závody práškové metalurgie Šumperk* [online]. Šumperk: obchodně technická služba a propagace s. p., 1990 [cit. 2020-12-07]. Dostupné z: <http://www.ferity.cz/termistory/negohm.pdf>
- [5] ŠPRINGL, Vít. Měření teploty: polovodičové odporové senzory teploty. *Vyvoj.hw.cz: teorie-a-praxe* [online]. 19. Srpen 2004 [cit. 2020-12-07]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/mereni-teploty-polovodicove-odporove-senzory-teploty.html>
- [6] *DS18B20 Programmable Resolution 1-Wire® Digital Thermometer: datasheet* [online]. [cit. 2020-12-07]. Dostupné z: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>
- [7] *Temperature and humidity module AM2302 Product Manual: datasheet* [online]. Aosong(Guangzhou) Electronics Co. [cit. 2020-12-07]. Dostupné z: <http://robotchip.ru/download/datasheet/AM2302-Datasheet.pdf>
- [8] HYRŠ, Jan: *Měření teploty termočlánky*. Brno, 2012. 32 s., CD. FSI VUT v Brně, Ústav strojírenské technologie, Odbor technologie tváření kovů a plastů. Vedoucí bakalářské práce Ing. Kamil Podaný, Ph.D.
- [9] *ThingSpeak™: Licensing FAQ* [online]. 2020 [cit. 2020-12-07]. Dostupné z: https://thingspeak.com/pages/license_faq
- [10] *Portál TMEP.cz: Mapa čidel* [online]. [cit. 2020-12-06]. Dostupné z: <https://tmep.cz/mapa/>
- [11] *Portál: myopenHAB* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://www.myopenhab.org>
- [12] ICS : 35.100.70 APPLICATION LAYER. *ISO/IEC 20922:2016: Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1*. 2016. 2016.
- [13] SHURE, Loren. THE MATHWORKS, INC. *Explore your IoT data with ThingSpeak and MATLAB: blog* [online]. 2016 [cit. 2020-12-07]. Dostupné z: <https://blogs.mathworks.com/loren/2016/03/23/explore-your-iot-data-with-thingspeak-and-matlab/>
- [14] *Portál: espressif.com* [online]. SHANGHAI: ESPRESSIF SYSTEMS, 2020 [cit. 2020-12-08]. Dostupné z: <https://www.espressif.com/en/products/socs/esp8266>
- [15] *Portál: EscapeQuotes* [online]. 2016 [cit. 2020-12-07]. Dostupné z: <https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>
- [16] *Portál: JLCPCB.COM* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://jlcpcb.com/>

- [17] Olovnaté a bezolovnaté pájky. *Portál: KONDIK.cz* [online]. 01.02.2019 [cit. 2020-12-07]. Dostupné z: <https://www.kondik.cz/olovnata-a-bezolovnata-pajka>
- [18] *Portál: itead.cc* [online]. [cit. 2020-12-08]. Dostupné z: https://www.itead.cc/media/wysiwyg/Products/ESP-01_Schematic.png
- [19] *16K I2C™ Serial EEPROM: 24AA16/24LC16B* [online]. 2012 [cit. 2021-5-30]. Dostupné z: <https://www.tme.eu/Document/7e02ac2588f5565e20bdaa8f1097e4ff/21703j.pdf>
- [20] *2C-bus specification and user manual: User manual* [online]. 2014 [cit. 2021-5-30]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [21] *PS Diagram* [online]. [cit. 2021-5-30]. Dostupné z: <https://app.diagrams.net/>
- [22] *Flowchart Maker & Online Diagram Software* [online]. [cit. 2021-5-30]. Dostupné z: <https://app.diagrams.net/>

SEZNAM SYMBOLŮ A ZKRATEK

IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
DPS	Deska plošných spojů
PTC	Positive Temperature Coefficient
NTC	Negative Temperature Coefficient
I2C	Inter-Integrated Circuit
TWI	Two Wire Interface
SPI	Serial Peripheral Interface
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
QoS	Quality of Service
EEPROM	Electrically Erasable Programmable Read-Only Memory
USB	Universal Serial Bus
UART	Universal asynchronous receiver-transmitter
GPIO	General Purpose Input Output
MCU	microcontroller
LED	Light-Emitting Diode
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TCP	Transmission Control Protocol
IEEE	Institute of Electrical and Electronics Engineers
RISC	Reduced Instruction Set Computer
SDA	Synchronous Data
SCL	Synchronous Clock
PLA	Kyselina polymléčná
ASCII	American Standard Code for Information Interchange
GND	Ground
STL	Standard Triangle Language
DNS	Domain Name System
mDNS	Multicast DNS
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
IDE	Integrated Development Environment

SEZNAM PŘÍLOH

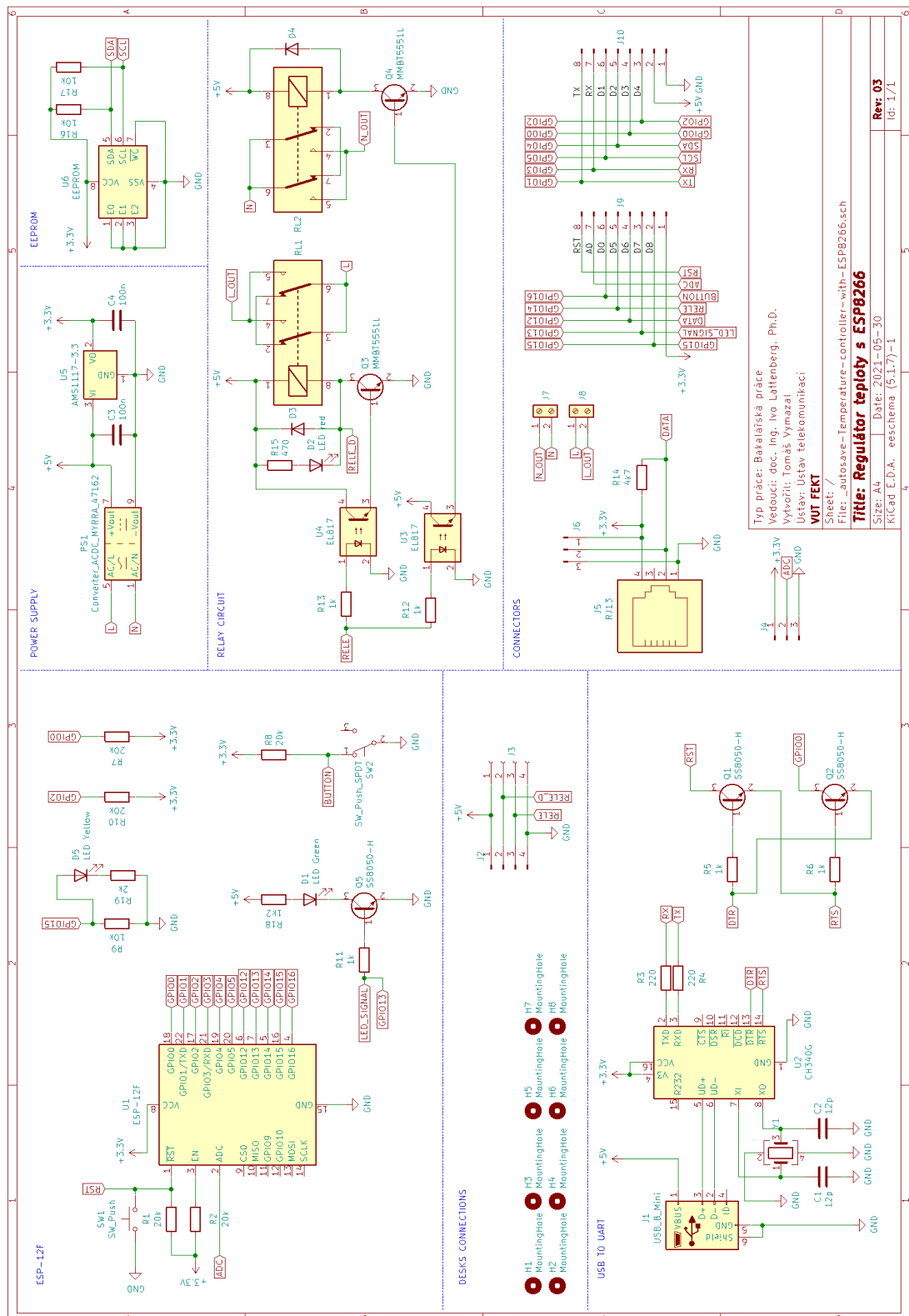
PŘÍLOHA A - SCHÉMA ZAPOJENÍ VYVÍJENÉ DPS 57

PŘÍLOHA B – Projekty programu KiCAD vyvíjených DPS jsou uloženy v příloženém archívu.

PŘÍLOHA C – Firmware je uložen v příloženém archívu.

PŘÍLOHA D – 3D modely, fotky, dokumenty k ověření regulace jsou uloženy v příloženém archívu.

Příloha A - Schéma zapojení vyvíjené DPS



OBSAH PŘÍLOŽENÉHO ARCHÍVU

PŘÍLOHA B

—	Temperature-controller-with-ESP8266 - verze 1
	logo_lib_vut.lib
	Temperature-controller-with-ESP8266-cache.lib
	Temperature-controller-with-ESP8266.kicad_pcb
	Temperature-controller-with-ESP8266.kicad_pcb-bak
	Temperature-controller-with-ESP8266.net
	Temperature-controller-with-ESP8266.pro
	Temperature-controller-with-ESP8266.sch
	Temperature-controller-with-ESP8266.sch-bak
	Temperature-controller-with-ESP8266.wrl
	Temperature-controller-with-ESP8266.xml
	test-cache.lib
	—Gerber
	Temperature-controller-with-ESP8266-B_Cu.gbr
	Temperature-controller-with-ESP8266-B_Mask.gbr
	Temperature-controller-with-ESP8266-B_SilkS.gbr
	Temperature-controller-with-ESP8266-Edge_Cuts.gbr
	Temperature-controller-with-ESP8266-F_Cu.gbr
	Temperature-controller-with-ESP8266-F_Mask.gbr
	Temperature-controller-with-ESP8266-F_SilkS.gbr
	Temperature-controller-with-ESP8266-job.gbrjob
	Temperature-controller-with-ESP8266.drl
—	Temperature-controller-with-ESP8266 - výpočetní deska verze 3
	logo_lib_vut.lib
	Temperature-controller-with-ESP8266-cache.lib
	Temperature-controller-with-ESP8266.kicad_pcb
	Temperature-controller-with-ESP8266.kicad_pcb-bak
	Temperature-controller-with-ESP8266.net
	Temperature-controller-with-ESP8266.pro
	Temperature-controller-with-ESP8266.sch
	Temperature-controller-with-ESP8266.sch-bak
	Temperature-controller-with-ESP8266.wrl
	Temperature-controller-with-ESP8266.xml
	test-cache.lib

—Gerber

Temperature-controller-with-ESP8266-B_Cu.gbr
Temperature-controller-with-ESP8266-B_Mask.gbr
Temperature-controller-with-ESP8266-B_SilkS.gbr
Temperature-controller-with-ESP8266-Edge_Cuts.gbr
Temperature-controller-with-ESP8266-F_Cu.gbr
Temperature-controller-with-ESP8266-F_Mask.gbr
Temperature-controller-with-ESP8266-F_SilkS.gbr
Temperature-controller-with-ESP8266-NPTH-drl_map.ps
Temperature-controller-with-ESP8266-NPTH.drl
Temperature-controller-with-ESP8266-PTH-drl_map.ps
Temperature-controller-with-ESP8266-PTH.drl

—Temperature-controller-with-ESP8266 - zdrojová deska verze 2

logo_lib_vut.lib
Temperature-controller-with-ESP8266-cache.lib
Temperature-controller-with-ESP8266.kicad_pcb
Temperature-controller-with-ESP8266.kicad_pcb-bak
Temperature-controller-with-ESP8266.net
Temperature-controller-with-ESP8266.pro
Temperature-controller-with-ESP8266.sch
Temperature-controller-with-ESP8266.sch-bak
Temperature-controller-with-ESP8266.wrl
Temperature-controller-with-ESP8266.xml
test-cache.lib

—Gerber

Temperature-controller-with-ESP8266-B_Cu.gbr
Temperature-controller-with-ESP8266-B_Mask.gbr
Temperature-controller-with-ESP8266-B_SilkS.gbr
Temperature-controller-with-ESP8266-drl_map.ps
Temperature-controller-with-ESP8266-Edge_Cuts.gbr
Temperature-controller-with-ESP8266-F_Cu.gbr
Temperature-controller-with-ESP8266-F_Mask.gbr
Temperature-controller-with-ESP8266-F_SilkS.gbr
Temperature-controller-with-ESP8266-job.gbrjob
Temperature-controller-with-ESP8266.drl

PŘÍLOHA C

└──FIRMWARE

- | Buffer.cpp
- | Buffer.h
- | Config.cpp
- | Config.h
- | Controller.cpp
- | Controller.h
- | EEPROM_24LC16B.cpp
- | EEPROM_24LC16B.h
- | LList.cpp
- | LList.h
- | main.ino
- | ProjectConfig.cpp
- | ProjectConfig.h
- | Timer.cpp
- | Timer.h
- | webFramework.cpp
- | webFramework.h
- | WebTextVariables.h
- | WifiModes.cpp
- | WifiModes.h

└──data

- config.html
- cross.png
- flag_0.png
- flag_1.png
- icon.png
- index.html
- javaScript.js
- style.css

PŘÍLOHA D

—DÍLY

47162.SLDPRT

47162.wrl

DISTAČNI_PODLOŽKA.SLDPRT

DISTAČNI_PODLOŽKA.STL

krabíčka.SLDPRT

krabíčka.STL

—FOTOGRAFIE

boční pohled 1.jpg

boční pohled 2.jpg

přední pohled 1.jpg

přední pohled 2.jpg

—OVĚŘENÍ REGULACE

Ověření regulace.xlsx

—PS DIGRAM

main.png

main.psdigram